МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з курсу «Комп'ютерні мережі»

для студентів спеціальностей
124 – Системний аналіз, 186 – Видавництво та поліграфія

Харків
НТУ ХПІ
2018

Методичні вказівки до лабораторної робот з курсу «Комп’ютерні мережі» для студентів спеціальностей 124 – Системний аналіз, 186 – Видавництво та поліграфія / Уклад. Ю. С. Шахновський. – Х. : НТУ «ХПІ», 2018. – 60 с.

Укладач Ю. С. Шахновський

Рецензент Л. М. Любчик

Кафедра системного аналізу та інформаційно–аналітичних технологій

Вступ

Цей курс присвячений вивченню мереж ЕОМ, галузі, яка увійшла в життя та працю усіх сфер. Курс містить 8 лабораторних робот. Перша частина цього курсу (лабораторні роботи 1– 4) присвячені вивченню засобів передачі пакетів у мережі, а друга частина (лабораторні роботи 5 – 8) найбільш поширеним програмам, які користувачі застосовують у мережах.

Лабораторна робота 1

АПАРАТУРА МЕРЕЖІ ETHERNET

Мета роботи: вивчення апаратури, використовуваної при побудові локальної мережі.

1.1. Загальні відомості

Найбільшого поширення серед стандартних мереж набула мережа Ethernet. Вперше вона з'явилася в 1972 році. У 1985 році мережа Ethernet стала міжнародним стандартом, її прийняли найбільші міжнародні організації за стандартами: комітет 802 IEEE (Institute of Electrical and Electronic Engineers) і ECMA (European Computer Manufacturers Association).

Основні характеристики первинного стандарту IEEE 802.3:

- топологія – шина;
- середовище передачі – коаксіальний кабель;
- швидкість передачі – 10 Мбіт/с;
- максимальна довжина мережі – 5 км;
- максимальна кількість абонентів – до 1024;
- довжина сегмента мережі – до 500 м;
- кількість абонентів на одному сегменті – до 100;
- метод доступу – CSMA/CD;
- передача вузько смугова, тобто без модуляції (моноканал).

Мережа Ethernet зараз найбільш популярна у світі, імовірно такою вона і залишиться найближчими роками. Цьому неабиякою мірою сприяло те, що з самого початку характеристики, параметри, протоколи мережі були відкриті, внаслідок чого величезне число виробників у всьому світі стали випускати апаратуру Ethernet, повністю сумісну між собою.

Під Ethernet розуміють мережу, в якій всі пристрої можуть чути інші пристрої. Для того щоб було зрозуміло, кому ж призначені передаванні в мережу дані, кожен пристрій в мережі має свій логічний номер. Якщо пристрій бачить пакет даних, який призначений саме для нього, то він спокійно його приймає, а інші пристрої ігнорують передачу. Якщо два (або більше) пристрої одночасно починають передавати дані, то всі вони замовкають і відновлюють активність через невизначений проміжок часу. Таким простим способом усуваються можливі конфлікти. Недоліком методу є те, що при великому навантаженні на мережу її продуктивність сильно знижується через часті конфлікти. У зв'язку з простотою конфігурації і дешевизною при досить хорошій продуктивності ethernet-мережі набули широкого розповсюдження. Очевидно, що Ethernet не призначений для створення розподілених мереж, його прерогатива – домашні/офісні або не дуже великі корпоративні мережі, які прокладаються в одній будівлі або декількох, якщо останні розташовані досить компактно, що пов'язано з невеликою максимальною довжиною мережного каналу.

У класичній мережі Ethernet застосовувався коаксіальний кабель на 50 ом двох видів (товстий і тонкий). Проте пізніше найбільшого поширення набула версія Ethernet, що використовує як середовище передачі витої пари. Визначений також стандарт для вживання в мережі оптоволоконного кабелю. Для обліку цих змін у початковий стандарт IEEE 802.3 були зроблені відповідні додавання.

Найстарішими мережами, побудованими за технологією Ethernet, є мережі на основі коаксіального кабелю. Коаксіальний кабель складається з центрального досить товстого провідника (одножильного або

багатожильного) і зовнішнього екрануючого обплетення. У ціх мережах застосовується коаксіальний кабель з хвилевим опором 50 ом. Хвилевий опір – відношення напруги до сили струму в даному перетинанню передавальної лінії при поширенні в ній електромагнітних коливань. Коаксіальний кабель зазвичай називають ethernet-кабелем або просто Ethernet, оскільки в комп'ютерній техніці він, як правило, використовується в основному для побудови мереж. Існує два варіанти організації мереж на коаксіальному кабелі: на товстому Ethernet і на тонкому Ethernet.

Приклад Ethernet на товстому коаксіальному кабелі показаний на рис.1.1

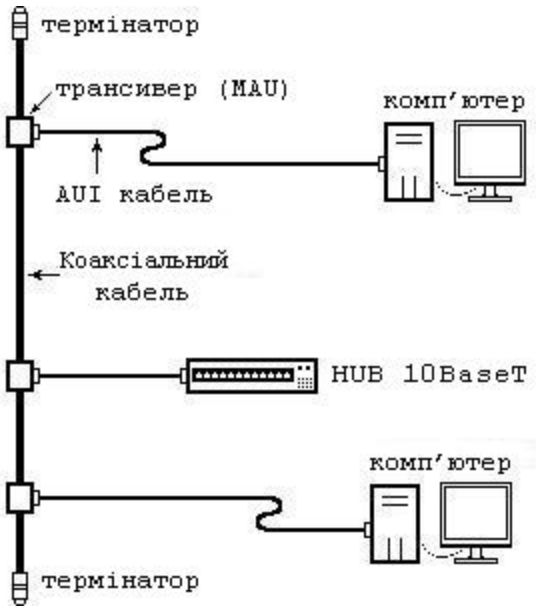


Рисунок. 1.1 – Ethernet на товстому коаксіальному кабелі

Стандарт також називається IEEE 10Base5. З рис 1.1 видно, що використовується шинна топологія, тобто пристрої навісили на кабель немов лампочки на ялинковій гірлянді. Через своє характерно забарвлення товстий Ethernet також називають жовтим кабелем. Він дозволяє використовувати сегмент завдовжки до 500 м при мінімальній відстані між точками підключення 2.5 метри (максимальна кількість точок підключення до сегмента дорівнює 100, а кількість сегментів мережі не може перевищувати 5 штук; детальніше про те, що таке сегмент і навіщо потрібне таке поняття,

буде розповідатися нижче). Пристрої підключаються до мережі за допомогою встановлюваного на кабель трансивера (MAU, Media Access Unit). Максимальна довжина кабелю між трансивером і пристроєм обмежена 25 м. При підключенні використовується 15-контактне рознимання AUI. Стандарт уже давно (в основному через дорожнечі відповідних кабелів і трансиверів) застарілий, і у даний час неможливо знайти у продажу нове обладнання для побудови мережі за цим стандартом, та і вже побудовані мережі є дуже великою рідкістю. Приклад Ethernet на тонкому коаксіальному кабелі показаний на рис. 1.2.

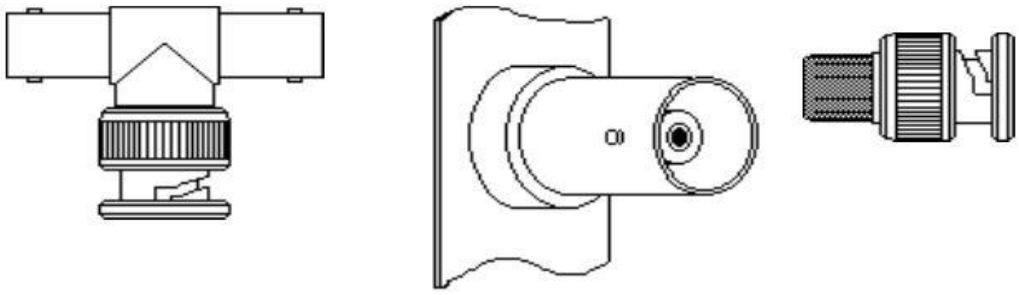


Рисунок 1.2 – Ethernet на тонкому коаксіальному кабелі

Цей сучасніший стандарт, який правильно називати 10Base2, аналогічний попередньому, але в ньому вже використовується дешевший тонкий кабель, а також відпадає необхідність в трансиверах. Тому такі мережі набули набагато ширшій популярності, ніж мережі на товстому кабелі. Пристрій підключається безпосередньо до мережі за допомогою Т-подібного конектора, де ніжка букви «Т» повернена до самого пристрою (наприклад, мережевої карти), а інші два кінці з'єднують відрізки кабелю і як би є його частиною. Можна вийняти конектор з пристрою, працездатність мережі від цього не погіршиться. Розняття на пристрою зазвичай позначається в тексті як BNC. Вони бувають, таки що накручуються, обжимні, або такі, що припаюються. Найбільш легкі в установці і надійні ті, що накручуються, але вони, на жаль, рідко коли є у продажу. Ті що

припаюються, дуже незручно встановлювати, особливо якщо установник перший раз в житті тримає в руках паяльник, тому найчастіше застосовуються обжимні конектори, які, до речі, теж досить надійні. Максимальна довжина сегмента 185 метрів, мінімальна відстань між точками підключення 0,5 метра, максимальна кількість точок підключення до сегменту становіть 30 при не більше ніж п’яти сегментів у мережі. Як видно з наведених характеристик мережі, заміна кабелю на дешевший обернулася зменшенням довжини сегменту мало не в три рази. Хоча мережа, як правило, може працювати при довжині сегмента до 200 м, все одно це куди менше, ніж 500.

Усі описані стандарти (як на товстому, так і на тонкому кабелі), обов’язково передбачають термінування з обох кінців. Це робиться за допомогою спеціальних наконечників, які встановлюються на кінці мережі. В принципі, можна обійтися і без термінування, але при цьому характеристики лінії різко погіршаться, що тут же в негативно вплине на продуктивність і надійність мережі. Один і лише один з термінаторів (тобто кінців мережі) має бути заземлений.

Вище згадувалося таке поняття, як сегмент. Сегмент – це ділянка мережі між двома репітерами (repeater), тобто пристроями, які підсилюють сигнал і тим самим дозволяють збільшувати довжину мережі. Репітер може мати як один, так і декілька мережних портів. Основна функція репітера – отримавши дані на одному з портів, негайно перенаправити їх на інші порти. В процесі передачі даних на інші порти дані формуються заново, щоб виключити будь-які відхилення, які могли виникнути під час руху сигналу від джерела. Репітери також можуть виконувати функцію, звану розділенням. Якщо репітер визначає велику кількість колізій, що відбуваються на одному з портів, він робить висновок, що сталася аварія десь на цьому сегменті, і ізолює його від іншої мережі. Ця функція була зроблена для запобігання поширенню помилок одного сегмента на всю мережу.

У репітерів є негативна риса, яка полягає в тому, що вони вносять затримку в поширення сигналу по мережі. Всі мережі Ethernet використовують протокол доступу, званий CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Щоб цей протокол працював нормально, йому необхідно мати можливість визначати виникнення колізії. CSMA/CD визначає це виникнення, порівнюючи дані, що знаходяться в мережі, з тими, що були відправлені до мережі. Якщо визначається відмінність, то це означає, що сталася колізія (одночасна передача двома пристроями), і передача негайно припиняється. Пристрій, як уже говорилося, потім чекає випадковий відрізок часу і повторює спробу передачі. Існує вада в CSMA/CD, яка обмежує розмір мережі. Послані біти не потрапляють миттєво у всі точки мережі, необхідний деякий відрізок часу, для того щоб сигнал пройшов по дротах і через кожен репітер у мережі. Цей час може бути виміряний, і він називається затримкою поширення (Propagation Delay). Якщо затримка поширення між джерелом сигналу і найбільш віддаленим джерелом мережі більша, ніж половина розміру найменшого пакета (frame), який може існувати, тоді CSMA/CD не зможе правильно визначити колізію, і дані в мережі можуть бути втрачені або спотворені. Тому не варто без особливої необхідності сильно захоплюватися репітерами, оскільки вони уповільнюють роботу мережі.

Згідно з проведеними розробниками Ethernet обчисленнями і вимірами, на шляху сигналу в мережі може бути не більше 4 репітерів і не більше 5 сегментів, причому лише до трьох з них можуть бути підключені пристрої. Ці висовки зазвичай виражаються у вигляді правила «5-4-3». Звичайно, в цілому в мережі може бути більше 4 репітерів (їх можна, в принципі, наставити скільки завгодно), але нас цікавить лише їх кількість між двома будь-якими пристроями. Це обмеження дає максимальну довжину мережі для товстого Ethernet, що дорівнює 2500 м і трохи менше кілометра для мережі на основі тонкого Ethernet.

Недоліки мереж, побудованих на коаксіальному кабелі пов’язані, по-перше, з невисокою пропускною спроможністю мережі, що дорівнює 10 Mbps (на практиці 7–8 Mbps), звідки і число «10» в назві. По-друге, шинна топологія не є надійною, оскільки розривання дротів призводить до розриву всієї мережі, та і використовувати її не завжди зручно.

Альтернатива коаксіальному кабелю – вита пара. Приклад сегмента, який застосовує виту пару показаний на рис. 1.3.

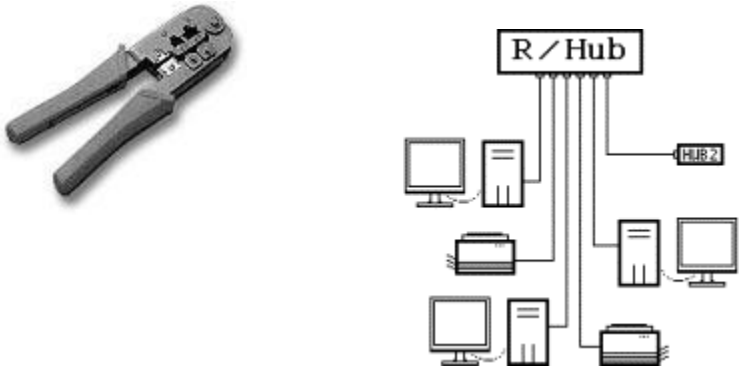


Рисунок 1.3 – Ethernet на витій парі

На відміну від мереж на коаксіальному кабелі, в Ethernet на витій парі використовується кабель, що складається з декількох скручених пар дротів (звідси і назва). У цей час подібні мережі найбільш поширені через їхню хорошу продуктивність, легкість у використанні і обслуговуванні, і при цьому досить низьку ціну як на кабель, так і на відповідне устаткування.

Центральним пристроєм у мережі є концентратор (хаб – Hub), до якого, власне, і підключається все інше мережне устаткування. Принцип роботи хаба дуже простий – він приймає від якого-небудь пристрою сигнал, підсилює його і передає по всіх напрямах (крім того, по якому він прийшов). Як видно з рисунка 1.3, використовується зіркоподібна топологія. Хаби можуть містити 4, 8, 16, 32 і так далі кількість портів, від цього залежить максимальна кількість мережних пристроїв, які можна до них підключити. Хаб зазвичай встановлюється на стіл, вішається на стіну або поміщається на

інше доступне місце, щоб було можливо швидко і без зусиль підключати до нього кабелі. Останні підключаються до хаба та інших мережних пристроїв за допомогою стандартного рознімання RJ-45, схожого на звичайние телефонне, але декілька більшого розміру (там використовується рознімання RJ-11). Якщо конектор виходить з ладу, то його зрізають разом з невеликим відрізком кабелю і ставлять новий. Конектори кріпляться за допомогою спеціального обжимного інструменту; існують також уже готові кабелі фіксованого розміру (наприклад, 5 або 3 метри), їх можна використовувати для підключення пристроїв, коли не потрібна велика віддаленість від концентратора. Одна пара з чотирьох у кабелі використовується на передачу, інша – на прийом. Як правило, концентратор має індикатори (світлодіоди) активності кожного порту.

Оскільки необхідні мережі, розмір яких перебільшує можливості одного сегмента, то потрібно вмити поєднувати сегменти поміж собою. Для витої пари можливо застосувати топології типу пасивна зірка і пасивне дерево. При цьому передбачається використання репітерів і репітерних концентраторів, що з'єднують між собою різні частини (сегменти) мережі. В результаті може сформуватися деревоподібна структура з сегментів різних типів.

Як сегмент (частина мережі) може виступати класична шина або одиничний абонент. Для шинних сегментів використовується коаксіальний кабель, а для променів пасивної зірки (для приєднання до концентратора одиночних комп'ютерів) – вита пара і оптоволоконний кабель. Головна вимога до отриманої в результаті топології – щоб в ній не було замкнутих доріг (петель). Фактично виходить, що всі абоненти з'єднані у фізичну шину, оскільки сигнал від кожного з них поширюється відразу у всі боки та не вертається назад (як у кільці). Максимальна довжина кабелю мережі в цілому (максимальний шлях сигналу) теоретично може досягати до 6,5 кілометрів, але практично не перевищує 3,5 кілометрів. Об'єднання сегментів мережі за допомогою репітерів та концентраторів показано на рис. 1.4.

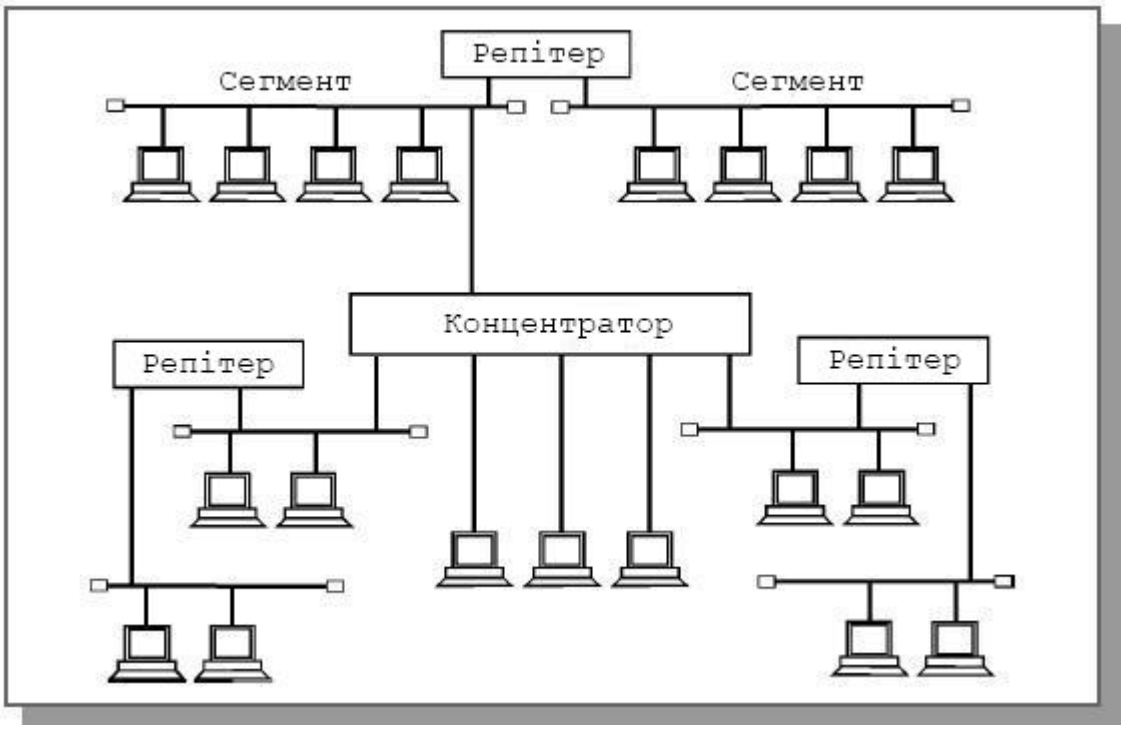


Рисунок 1.4 – Об’єднання сегментів мережі з електричною розв’язкою

Спосіб з’єднання сегментів, описаний вище, дає так звану електричну розв’язку. В цьому випадку з одного сегмента мережі передається на інший сигнал, але відсіюється перешкода. Недоліком такого способу з’єднання є передача сигналу у всі сегменти. Як у тих, в яких знаходиться приймач цього сигналу, так і в тих, в яких він даремний. Це не дозволить створювати мережі великого розміру. Наступне збільшення розміру мережі може дати з’єднання мереж з логічною розв’язкою між ними. Це досягається вживанням дорожчих пристроїв зв’язку між сегментами, званих switch, які вміють розрізняти, в який з сегментів призначений сигнал, і видавати його не у всі підключені сегменти, а лише в сегмент приймача. Зрозуміло, що switch буде дорожчим, ніж хаб або репітер. Другий недолік такого пристрою – велика затримка, яку він вносить до роботи мережі. Якщо в звичайному репітері сигнал, що прийшов, можна починати передавати в наступний сегмент відразу за приходом першого біта, то в пристрої з логічною розв’язкою необхідно зібрати весь пакет, а лише потім, на основі адреси приймача пакету можна визначити, в який з сегментів його потрібно передати.

Існує ще один спосіб зв'язку сегментів мережі між собою – через комп'ютер з двома мережними картами. Кожна мережна карта підключається до свого сегмента мережі. Пакет, що прийшов на одну з мережних карт, приймається і запам'ятовується в пам'яті комп'ютера. Потім спеціальне програмне забезпечення видає цей пакет через другу мережну карту в наступний суміжний сегмент. Комп'ютер, використовуваний в такому режимі, називають шлюзом. При такому з'єднанні мережі з пакетом можна виконати ще багато операцій за допомогою програм. Наприклад, підраховувати сумарний обсяг трафіку, що проходить через шлюз, для різних джерел даних. Або на основі вмісту пакета приймати рішення, відправляти його далі чи ні (фільтрація пакетів). Але програмна обробка вимагає часу і дає додаткову затримку.

1.2. Програма виконання роботи

1. Ознайомитися з пристроями, використовуваними при створенні мереж, і способами їх з'єднання.

2. Порівняти пристрої, вживані в мережі кафедри, з пристроями, які використані для підключення до Internet домашніх комп'ютерів.

3. Проаналізувати вартість домашнього підключення до мережі і використовувану технологію.

Контрольні запитання

1. Перерахувати порівняльні переваги і недоліки вивчених топологій з'єднання мереж.

2. У чому різниця між логічною і фізичною розв'язкою сегментів мережі?

3. Які додаткові переваги дає програмна розв'язка порівняно з логічною розв'язкою? Що є платою за ці переваги?

4. Які основні параметри характеризують функціональні можливості сегмента мережі?

5. Для чого в конструкцію дротів вводиться обплетення?

6. У чому основний недолік побудови мережі за допомогою топології загальної шини?

7. Які топології використовуються для побудови домашніх комп'ютерних мереж?

Лабораторна робота 2

НАЛАШТУВАННЯ МЕРЕЖІ НА ДОМАШНЬОМУ КОМПЮ'ТЕРІ

Мета роботи: навчитися налаштовувати мережу на комп'ютері, підключеному до локальної мережі, вивчити програми аналізу якості роботи мережі.

2.1. Загальні відомості

Для вивчення налаштування мережі нам будуть потрібні права адміністратора на даному комп'ютері. Вам потрібно почекати, поки буде активований вхід адміністратора, закінчити сеанс користувача stud і увійти до системи під ім'ям користувача stud1 з паролем stud.

Перейдемо у вікно налаштування мережі.

1. Натисніть кнопку **«Пуск»**

2. Оберіть «**Настройка"** >> **"Мережа та віддалений доступ к мережі»**

3. Відкриється вікно «**Мережа та віддалений доступ до мережі»**

4. Правою кнопкою миші клацніть по значку «**Підключення до локальної мережі»** та оберіть **«Властивості»**

5. Відкриється вікно налаштувань мережного підключення. У нім звернемо увагу на пункти:

а) «Клієнт для мереж Microsoft». Ця компонента дозволяє вам використовувати файли і принтери, які інші комп'ютери вашої локальної мережі зробили доступними для загального використання;

б) «Служба доступу до файлів і принтерів мережі Microsoft». Ця компонента дозволяє вам робити доступними для загального користування файли і принтери свого комп'ютера;

в) «планувальник пакетів QoS». Розподіляє смугу пропускання каналу передачі між мережними застосуваннями, що працюють на комп'ютері;

г) «Протокол Internet (TCP/IP)». Включає драйвер протоколів TCP/IP, які забезпечують мережний і канальний рівень передачі даних.

Пункти а) – в) налаштовувати не треба. Для налаштування пункту г) застосовуються числа, які ви повинні отримати у свого провайдера. Виберемо пункт Протокол Інтернет (TCP/IP) і увійдемо до його властивостей. Перша радіокнопка дозволить вибирати автоматичне або ручне налаштування параметрів. Якщо обрати автоматичне налаштування, то відбуватиметься запит даних для Internet параметрів по мережі. Це можливо лише в тому випадку, якщо у вашій локальній мережі є спеціальний сервер, відповідальний за автоматичне призначення IP–адрес. При виборі ручного налаштування вам доведеться ввести 3 числа, що визначають роботу протоколу IP для вашого комп'ютера:

а) це його унікальна IP–адреса, яка використовуватиметься іншими комп'ютерами як адреса приймача при передачі пакетів Вашому комп'ютеру;

б) маска підмережі. Вона використовується для визначення комп'ютерів, що знаходяться з вашим комп'ютером в одному сегменті мережі, яким можна передати пакет безпосередньо, не використовуючи проміжний шлюз.

в) основний шлюз. Це адреса комп'ютера, якому відсилатимуться ті пакети, які потрібно віддати комп'ютерам за межами поточного сегмента. Його завдання визначити подальший шлях цих пакетів у мережі.

Цих трьох чисел вистачає для використання TCP/IP і завдання адресів Інтернет у вигляді IP адресів. Для використання доменної служби імен також необхідно налаштувати службу DNS. Для цього необхідно ввести адресу основного і додаткового шлюзів. Два шлюзи необхідні для того, що якщо

пропаде зв’язок з першим з DNS сервером, то використовуватиметься другий.

Приклад налаштування параметрів TCP/IP показаний на рис 2.1.

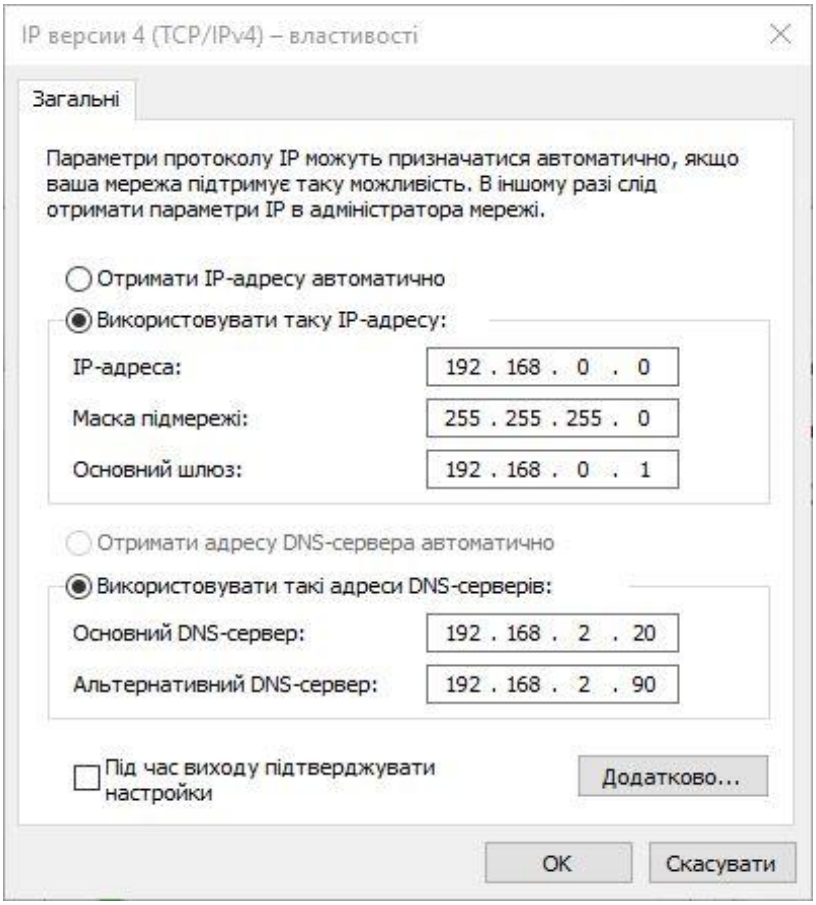


Рисунок 2.1 – Приклад налаштування параметрів TCP/IP

6. Натисніть кнопку «Додатково». У закладці «Параметри IP» Ви потрапляєте в меню, яке дає можливість налаштувати на комп’ютері більш ніж одну мережну карту. Для кожної мережної карти потрібно наладжувати свою IP адресу, маску підмережі і основний шлюз. Крім того, для роботи з декількома мережними картами, вам доведеться налаштувати роутингову таблицю, за допомогою якої драйвер IP визначає для кожного пакета через яку мережну карту і куди цей пакет посилати. У закладці DNS є можливість задати більше двох DNS серверів. Це підвищить надійність розпізнавання, але збільшить час відмови, яка станеться при зверненні до неправильної адреси, тому що час відмови складається з суми тайм-аутів усіх серверів, вказаних для DNS, тому не рекомендується виставляти більше 2 – 3 серверів. У закладці WINS є можливість налаштувати службу імен, альтернативну

DNS. У закладці «безпека» можна налаштувати службу фільтрації пакетів. Стандартний фільтр пакетів ОС Windows має обмежені можливості. Для використання як фільтр зручніше застосовувати спеціалізовані програми, наприклад Outpost. Потрібно пам'ятати, що помилка при налаштуванні фільтра пакетів може привести до того, що мережне програмне забезпечення працюватиме неправильно. Тому краще не зловживати цією можливістю до того, як ви будете в ній нормально розумітися.

7. Натиснемо двічі кнопку «Відміна». Повернемося у вікно «Підключення до локальної мережі, властивості». Поглянемо, які компоненти і драйвери можна включити на комп'ютері додатково, окрім вже включених. Для цього натискуватимемо клавішу «Установити». У нас з'являється вікно «Вибір мережного компоненту» з трьома пунктами «Клієнт», «Служба», «Протокол». У пункті «Клієнт» можна обрати програми, які дадуть вашому комп'ютеру можливість використовувати деякі додаткові послуги. У пункті «Служба» (правильніше, напевно було використовувати термін «сервер») можна обрати програми, установка яких дасть вашому комп'ютеру можливість надавати додаткові послуги для інших машин по мережі. У пункті «драйвер» у вас з'явиться можливість установити способи доставки пакетів, альтернативні стандартному TCP/IP:

а) виберемо пункт драйвер. При установці драйвера Мікрософт TCP/IP версії 6 буде використана інша система мережних адрес, чим чотирихбайтні адреси, які використовуються на комп'ютері зараз. Перехід на шестибайтні адреси допоможе вирішити проблему дефіциту IP–адресів, яку ми маємо зараз. Для переходу на 6-байтні адреси все готово. Для цього необхідно лише, щоб усі користувачі Інтернет замінили чотирихбайтні адреси на шестибайтні. Але, оскільки це необхідно робити всім користувачам Інтернет одночасно, то такий перехід не виходить організувати вже багато років.

Вибір пункту NWLink IPX/SPX/NetBios-сумісний транспортний протокол встановить на ваш комп'ютер драйвер протоколу IPX. Цей протокол забезпечує роботу на мережному і транспортному рівні,

альтернативну протоколу TCP/IP. IPX дає швидкість передачі даних вищу, ніж при використанні TCP/IP. Але в IPX відсутня властивість налаштування масштабів. При передачі пакетів з використанням цього протоколу через велике число шлюзів виникають проблеми. Тому протокол IPX програв TCP/IP при виборі протоколу для побудови світової мережі. Вже будучи стандартом при побудові глобальної мережі, TCP/IP витіснив IPX і з локальних мереж, оскільки локальні мережі вважають за краще будувати на принципі Intranet, коли програмне забезпечення повністю відповідає програмам, використовуваним для глобальної мережі. Але для роботи деяких старих програм все ще може бути потрібне установка на вашому комп'ютері драйвера IPX;

б) натиснемо «відміна» і виберемо пункт «клієнт». Установлення компоненти «Клієнт для мереж Netware» дасть можливість нашому комп'ютеру підключатися до файлових серверів, що працюють під управлінням ОС, розробленою фірмою Netware. Файловим сервером називають машини, призначені для зберігання і передачі по мережі великої кількості фалів. Основна вимога до таких машин – швидкість, з якою можна одночасно скачувати файли. Сервери Netware розв'язують проблему швидкості краще за всіх. Але у них є інший недолік. Файлові сервери Netware є виділеними. При виконанні функцій файлового сервера машина, що працює під управлінням операційної системи Netware, не може бути використана для інших функцій. Сервери, що працюють під операційними системами Unix і Windows, хоч і поступаються серверам Netware за швидкістю передачами файлів по мережі, але дозволяють використовувати обчислювальну машину для інших функцій паралельно роботі в режимі файл сервера. Тому в сучасних мережах частіше використовують саме їх. А сервери Netware стали рідкістю;

в) натиснемо «відміна» і виберемо пункт «Служба». Серед можливих компонент цього пункту нам може бути цікавий пункт «Персональний Web–

сервер». Установлення цієї компоненти дасть вашому комп'ютеру можливість працювати як Web–сервер.

8. Оберемо кнопку відміна 3 рази підряд і вийдемо з налаштування мережі без запам'ятовування.

9. Запустимо вікно командного рядка. Для цього натискуватимемо кнопку «Пуск», «Виконати», наберемо в рядку «cmd» і натискуватимемо кнопку «ок».

10. Виконаємо команду ipconfig /all. За допомогою цієї команди ми можемо поглянути IP і Ethernet адреси комп'ютера, за яким ми працюємо, а також інші налаштування мережі на комп'ютері.

2.1.1. Робота з ARP протоколом

1. Подамо команду ping 172.16.201.10. Детальніше, як працюватиме ця команда, ми вивчатимемо пізніше на цьому занятті. А поки потрібно знати, що вона посилає пакет на машину з вказаною адресою.

2. Подамо команду arp –a. Ми бачимо поточну таблицю ARP з відповідностями між IP і Ethernet адресами. Для додавання в неї записів нам необхідно обмінюватися пакетами з іншими машинами.

3. За допомогою команди ping організуйте обмін пакетами з іншими машинами в аудиторії і доможіться, щоб записи про всіх них були в ARP таблиці.

4. Почекавши декілька хвилин, знову погляньте на ARP–таблицю. Зверніть увагу, що записи з неї пропали. Це пов'язано з динамічним характером цих записів. Вони заносяться в ARP таблицю не назавжди, а лише на невеликий час. Якщо в перебігу цього часу запис використовується, то час її життя подовжується. Якщо звернень до цього запису немає, то вона через деякий проміжок знищується.

Яка користь від такого способу зберігання записів в ARP–таблиці? Якщо який-небудь комп'ютер буде вимкнений або у нього буде замінена мережна карта на карту з іншою Ethernet адресою, то ARP–таблиця буде

містить неправильний рядок. При статичному способі зберігання рядків виправляти цю помилку повинен системний адміністратор. Причому цю роботу необхідно виконати на всіх комп'ютерах сегмента. Замість цього використовується динамічний спосіб зберігання. Рядок, який не використовується декілька хвилин, відкидається. При новому зверненні до IP–адреси з цього рядка ARP таблиця заповнюється заново. Оскільки вміст рядка, до якого йдуть часті звернення, зберігається, то мережа не перевантажується ARP запитами. Цей метод дозволяє підтримувати правильну таблицю ARP без втручання адміністратора.

5. Надішлемо пакет за адресою 172.16.201.1 за допомогою команди ping. Наберемо arp –a. Ми побачимо, яка Ethernet адреса відповідає цій IP адресі. За допомогою команди «arp –s Ip_addr Ehetnet_addr» занесемо цю відповідність в ARP таблицю. Спробуємо зв'язатися з комп'ютером 172.16.201.1. Обмін пакетами можливий. Тепер видалимо введений нами рядок командою "arp –d IP_addr", і введемо її заново, але задавши Ethernet адресу неправильно. Після цього спробуємо зв'язатися з видаленим комп'ютером знову. Побачимо, що обмін пакетами при неправильний ARP–таблиці неможливий. Необхідно видалити введений статичний рядок ARP таблиці, тому що статичні рядки не зникають з часом і навіть зберігаються при перезавантаженні комп'ютера.

2.2.2. Команда ping

1. Розберемося з командою ping, яка вже використовувалася для обміну пакетами між комп'ютерами. Ця команда призначена для перевірки якості роботи IP–мереж. Вона відправляє запити (ICMP Echo-request) протоколу ICMP вказаному вузлу мережі і фіксує відповіді, що надходять (ICMP Echo-Reply). Час між відправкою запиту і отриманням відповіді (RTT, від англ. Round Trip Time) дозволяє визначати двосторонні затримки по маршруту і частоту втрати пакетів. З його допомогою можна побічно визначати завантаженість на каналах передачі даних і проміжних пристроях.

Також пінгом часто помилково називають час, витрачений на передачу пакета інформації в комп'ютерних мережах від клієнта до сервера і назад, від сервера до клієнта. Цей час називається лагом (англ. відставання; затримка, запізнювання) або власне затримкою і вимірюється в мілісекундах. Лаг зв'язаний із швидкістю з'єднання і завантаженістю каналів протягом всього шлюху від клієнта до сервера.

Повна відсутність ICMP-відповідей може також означати, що видалений вузол (або який-небудь з проміжних маршрутизаторів) ICMP Echo-reply або ігнорує ICMP Echo-request. Програма ping є одним з основних діагностичних засобів у мережах IP і входить до постачання всіх сучасних мережних операційних мереж.

2. Пошлемо команду ping за адресою 172.16.201.5. Ми можемо спостерігати якість зв'язку з цим вузлом. Вона ідеальна, вузол знаходиться на одному сегменті з нашим комп'ютером. Якщо послати ping за адресою 172.17.10.2, то час відповіді буде більшим, оскільки цей комп'ютер знаходиться в іншому корпусі, через кілька сегментів мережі.

3. Наберемо команду ping без параметрів. Ми побачимо на екрані допомогу, яка описує роботу утиліти. Зверніть увагу на ключі. Ключ -t задає обмін не 4-х пакетів, як за замовчуванням, а нескінченного числа пакетів. Запити ping будуть надсилатися доти, поки ви не зупините обмін, натиснувши Ctrl+C. При цьому ви можете оцінити відсоток пакетів, які губляться. Ключ «-w число» дозволить задати час, в перебігу якого відбувається очікування відповіді на запит. Якщо відповідь прийшла з затримкою, більшою зазначеної в цьому ключі, то вважається, що відповіді не було.

2.1.3. Команда traceroute

1. traceroute – це службова комп'ютерна програма, призначена для визначення маршрутів прямування даних в IP–мережах. traceroute зоснована на протоколі ICMP.

Програма traceroute виконує відправку даних указаному вузлу мережі, при цьому відображаючи відомості про всіх проміжних маршрутизаторів, через які пройшли дані на шляху до цільового вузла. У разі проблем при доставці даних до якогось вузла програма дозволяє визначити, на якій саме ділянці мережі виникли неполадки. Тут хочеться відзначити, що програма працює тільки в напрямку від джерела пакетів і є досить грубим інструментом для виявлення неполадок у мережі. У силу особливостей роботи протоколів маршрутизації в мережі Інтернет, зворотні маршрути часто не збігаються з прямими, причому це справедливо для всіх проміжних вузлів. Тому, ICMP відповідь від кожного проміжного вузла може йти своїм власним маршрутом, загубитися або прийти з великою затримкою, хоча в реальності з пакетами, які адресовані кінцевому вузлу, цього не відбувається. Крім того, на проміжних маршрутизаторах часто стоїть обмеження числа відповідей ICMP в одиницю часу, що призводить до появи помилкових втрат. traceroute входить у поставку більшості сучасних мережних операційних систем. У системах Microsoft Windows ця програма має назву tracert, а в системах Unix, Cisco і Mac OS – traceroute.

2. Принцип роботи traceroute

Для визначення проміжних маршрутизаторів traceroute відправляє серію (зазвичай три) пакетів даних цільового вузла, при цьому кожного разу збільшуючи на 1 значення поля TTL («час життя»). Це поле зазвичай вказує максимальну кількість маршрутизаторів, яка може бути пройдена пакетом. Перша серія пакетів відправляється з TTL, рівним 1, і тому перший же маршрутизатор повертає назад повідомлення ICMP, яке вказує на неможливість доставки даних. traceroute фіксує адресу маршрутизатора, а також час між відправленням пакета і отриманням відповіді (ці відомості виводяться на монітор комп’ютера). Потім traceroute повторює відправку серії пакетів, але вже з TTL, рівним 2, що дозволяє першому маршрутизатору пропустити їх далі.

Процес повторюється доти, поки при певному значенні TTL пакет не досягає цільового вузла. При отриманні відповіді від цього вузла процес трасування вважається завершеним.

На цільовому хості IP-дейтаграма з TTL = 1 не відкидається і не викликає ICMP-повідомлення типу термін закінчився, а повинна бути віддана додатку. Повідомлення про помилку досягається таким чином: traceroute дейтаграми, які відсилаються, містять UDP-пакет з таким номером UDP-порта адресата (що перевищує 30 000), що він свідомо не використовується на хості, якій адресується. В пункті призначення UDP-модуль, отримуючи подібні дейтаграми, повертає ICMP-повідомлення про помилку «порт недоступний». Таким чином, щоб дізнатися про завершення роботи програмою traceroute досить виявити, що надійшло ICMP-повідомлення про помилку цього типу.

3. Подамо команду tracert -d 172.17.10.2. Побачимо, що цільовий комп'ютер досягається через кілька проміжних шлюзів. При цьому для кожного шлюзу ми побачимо 3 результата вимірювання RTT, що дозволить нам судити про якість зв'язку не тільки з цільовим вузлом, а й про якість зв'язку з кожним проміжним вузлом.

4. Подамо команду tracert без параметрів. Ми побачимо список ключів, які можна використовувати з цією командою. Зверніть увагу на ключ -d – його використання має сенс, якщо є проблеми з роботою nameserver або якщо є бажання заощадити час, необхідний для переведення адрес в імена. Ключ -h дозволяє задати максимальну кількість вузлів, при якій буде тривати пошук цільової машини. Задане в ньому число обмежує TTL, з яким будуть відправлятися пакети. За умовчанням це число дорівнює 30. В результаті можна дістатися до комп'ютерів, віддалених від початкового на 29 шлюзів. Це обмеження запобігає ситуації, коли в результаті помилки адміністратора в мережі виникає «цикл», в якому два комп'ютери віддають пакети один одному. І при нарощуванні TTL пакети все довше крутяться між ними.

2.2. Програма виконання роботи

1. Ознайомитись зі способами додавання нових і видалення старих компонент в мережному забезпеченні комп'ютерів, які використовують ОС WINDOWS.

2. Ознайомитися з налаштуванням підключення комп'ютера до Internet.

3. Дослідити ARP таблицю робочого комп'ютера. Провести експерименти: що відбувається із зв'язком при зміні ARP таблиці.

4. Навчитися використовувати команди аналізу якості роботи мережі – ping и tracert.

Контрольні запитання

1. У чому різниця між програмою клієнтом і програмою сервером?

2. Які сервери встановлені на комп'ютерах кафедри?

3. З якими протоколами ви познайомилися на лабораторній роботі?

4. Які дані повинні вам бути відомі перед початком налаштування IP протоколу. Звідки ви їх дізнаєтеся?

5. Як налаштувати IP на машині з декількома мережними картами.

6. Для чого потрібно налаштовувати DNS?

7. Які переваги дає використання налаштування протоколу IP за допомогою DHCP?

8. У чому різниця між статичними і динамічними рядками в ARP–таблиці? Які переваги дає використання динамічних записів в ARP таблиці?

9. Що вимірює команда ping?

10. Які проблеми виникають при спробі виміряти час передачі сигналів в один бік?

11. Яку додаткову інформацію дає команда tracert у порівнянні з ping?

12. Яке максимальне число шлюзів між комп'ютерами має бути в Internet? Які проблеми виникнуть, якщо їх буде більше?

Лабораторна робота 3

ДОСЛИДЖЕННЯ ТОПОЛОГІЇ МЕРЕЖІ УНІВЕРСИТЕТУ

Мета роботи: навчитися визначати топологію мережі з допомогою команди tracert. Вивчити реакцію команди tracert на нестандартні ситуації в мережі.

3.1. Загальні відомості

Мережа університету складається з сегментів, сполучених між собою через шлюзи. Кожному сегменту відповідає діапазон IP–адрес, що йдуть підряд. Його можна задати початковою і кінцевою адресами. А можна початковою адресою і маскою. Маска визначає кількість адрес у сегменті, і задає кінцеву адресу побічно.

Належність адреси x сегменту з початковою адресою d (destination) і маскою m (mask) можна перевірити формулою if((x & m) == d).

Необхідно визначити ознаку належності двох адрес одному сегменту при експериментальному дослідженні мережі. Виконуємо команду tracert для адрес x1 і x2. Якщо при виконанні tracert для x1 і x2 збігаються всі проміжні шлюзи і збігається кількість стрибків до досягнення кінцевого пункту, то ці дві адреси належать одному сегменту. Якщо не збігаються – різним.

Приклад виконання команди tracert для адрес 172.17.56.1 і 172.17.56.2, які належать одному сегменту:

I) `tracert –d 172.17.56.1`

```
трасування маршруту до 172.17.56.1
1    1 ms    1 ms    1 ms  172.16.201.1
2    1 ms    1 ms    1 ms  172.17.56.1
 Трасування завершене.
```

II) `tracert –d 172.17.56.2`

```
трасування маршруту до 172.17.56.1
1    1 ms    1 ms    1 ms  172.16.201.1
2    1 ms    1 ms    1 ms  172.17.56.2
```

Трасування завершене.

Бачимо, що в цих двох експериментах шлях сигналу проходить через шлюз 172.16.201.1 і потім закінчується на цільовій машині команди. Це ознака того, що адреси 172.17.56.1 і 172.17.56.2 належать одному сегменту.

Для адреси 172.16.201.10 шлях, виведений на екран командою tracert, буде інший.

III) `tracert -d 172.16.201.10`

```
трасування маршруту до 172.16.201.10
1    1 ms    1 ms    1 ms  172.16.201.10
```

Цільова адреса досягнута в один стрибок, і це дозволяє вважати, що комп'ютер з цією адресою знаходиться в іншому сегменті, ніж два попередніх.

Далі необхідно експериментально визначити межи сегменту. Для цього потрібно знайти дві адреси, що йдуть одна за іншою, які дають різні шляхи проходження tracert. Це і буде ознакою, що молодша адреса – кінець одного сегмента, а наступна за ним адреса – вже початок іншого. Такий експеримент ускладнюється тим, що перша і остання адреса кожного сегмента використовується по-особливому. Перша адреса IP, що належить сегменту, задає адресу, яку називають адресою цього сегмента. Остання IP адреса сегменту задає броадкаст, тобто звернення до всіх машин сегменту відразу. На звернення за такою адресою відповідають всі машини сегмента. Але на екран команда tracert виводить результати відповіді лише від того комп'ютера, сигнал від якого прийшов перший. Ми бачимо, що адреса сегмента і броадкаст використовуються по-особливому, їм не відповідають реальні машини, і число стрибків для досягнення цієї адреси може не збігатися (воно рівне або на один менше), чим адреси реальних комп'ютерів сегмента. Тому при знаходженні меж сегмента потрібно використовувати не його першу і останню адреси, а другу і передостанню. Так, для сегмента з межами 1.1.1.0 – 1.1.1.225 ми зможемо виявити ці кордони, якщо поставимо такі експерименти:

1) Виконаємо команду tracert для 1.1.1.1 (друга адреса сегменту).

2) Виконаємо команду tracert для 1.1.0.224 (передостання адреса попереднього сегмента).

3) Виконаємо команду tracert 1.1.1.224 (передостання адреса для досліджуваного сегмента).

4) Виконаємо команду tracert 1.1.2.1 (друга адреса сегмента наступного за досліджуваним).

Якщо перший і другий експерименти дали різні шляхи проходження tracert, третій і четвертий експеримент також дали різні шляхи, а всі виконання tracert для адрес 1.1.1.1 – 1.1.1.224 дають один маршрут, що повторюється, то це означає, що адреси 1.1.1.0 – 1.1.1.225 належать одному сегменту, з destination 1.1.1.0, маскою 255.255.255.0 та broadcast 1.1.1.255.

Введемо графічні позначення для завдання зв'язків елементів мережі між собою. Ці позначення показані на рис. 3.1 та 3.2

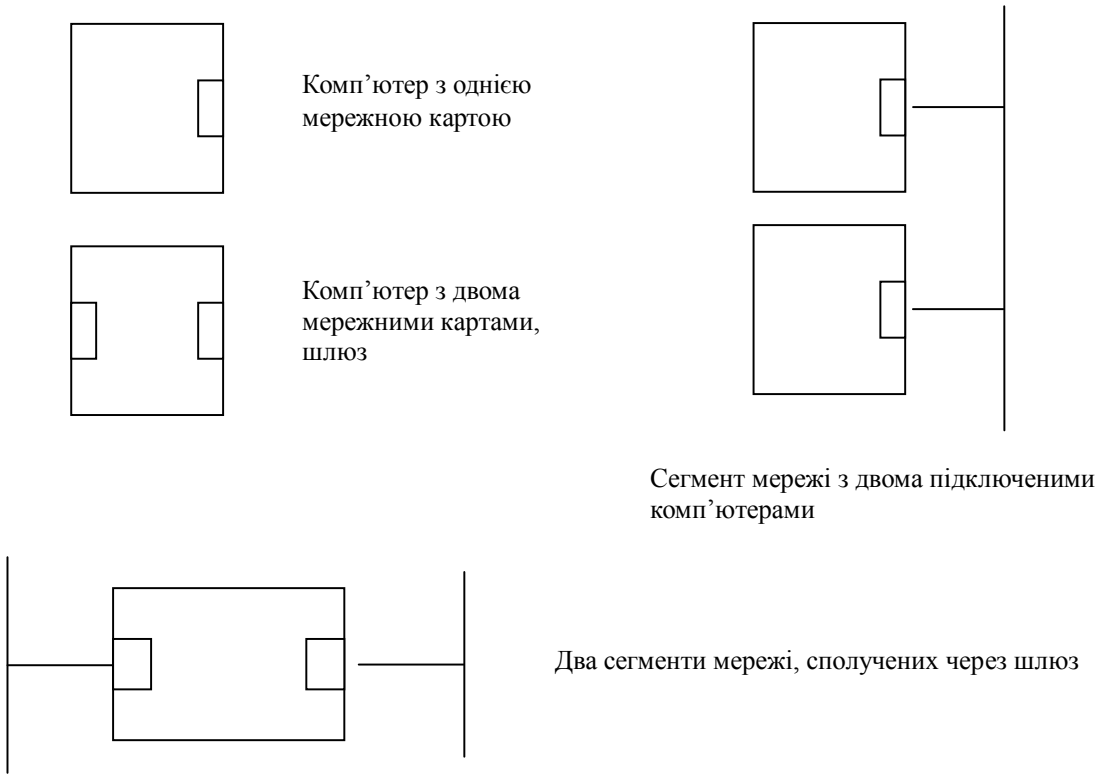


Рисунок 3.1 – Позначки для зображення комп'ютерів та їх з'єднання між собою

Якщо необхідно показати IP адресу, присвоєну мережній карті комп'ютера, і діапазон адрес, присвоєних сегменту, то графічно це зображується так, як на рис 3.2.

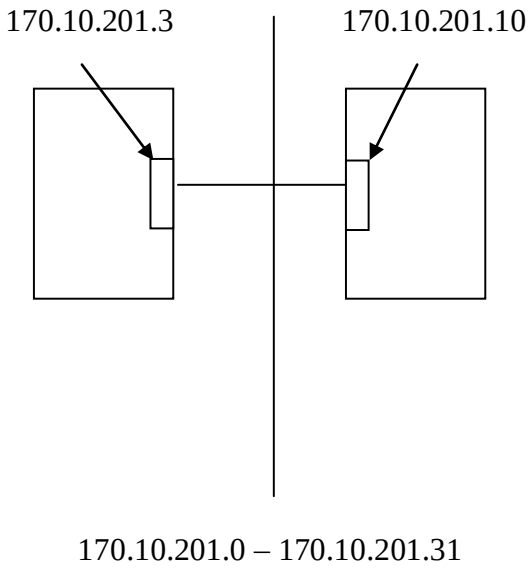


Рисунок 3.2 – Приклад зображення діапазону адрес сегмента

Тепер проведемо аналіз результатів виконання команди

```
tracert -d 194.44.235.1
трасування маршруту до 194.44.235.1
1    1 ms    1 ms    1 ms  172.16.201.1
2    1 ms    1 ms    1 ms  172.17.56.1
3    1 ms    1 ms    1 ms  192.168.12.1
4    1 ms    1 ms    1 ms  194.44.235.1
 Трасування завершене.
```

Зобразимо результати аналізу цього експерименту на рис. 3.3.

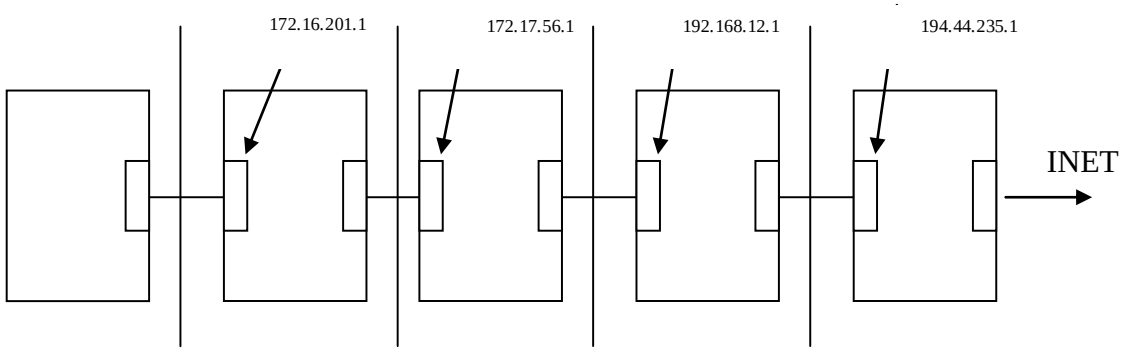


Рисунок 3.3 – Наявне зображення результатів експерименту виконання команди tracert

Команда tracert показує нам IP адреси «ближніх» мережних карт шлюзів, по шляху передачі пакета.

Потім необхідно взнати адресу «дальньої» мережної карти шлюзу. Відомо, що всі мережні карти одного сегменту досягаються в одну й теж кількість стрибків. Нехай це N. Тоді «ближня» мережна карта шлюзу, через який пакет потрапляє на цей сегмент буде досягнута в (N-1) стрибків. І в стільки ж стрибків (N-1) ми отримаємо відповідь, якщо пошлемо tracert безпосередньо на далеку мережну карту шлюзу. Цей аналіз відображений на рис. 3.4.

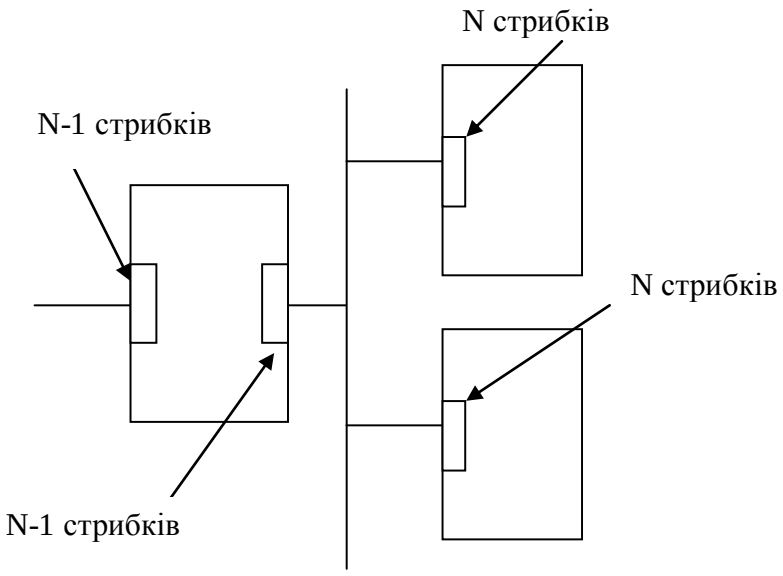


Рисунок 3.4 – Пошук дальньої мережної карти

Тобто для того, щоб дізнатися IP адресу дальньої мережної карти, нам необхідно послідовно посилати команду tracert на адреси, що належать сегменту. Окрім першої і останньої. Якщо ми знайдемо IP адресу, яка досягається на один стрибок швидше, ніж останні адреси цього сегмента, то це ознака того, що це IP адреса задньої мережної карти шлюзу, через яку пакети потрапляють на досліджуваний сегмент.

Отримуємо такий алгоритм знаходження дальньої адреси мережної карти.

1. Знаходимо межи сегмента, якому належить дальня мережна карта.

2. Послідовно перебираємо всі адреси сегмента, з метою знайти адресу, яка досягається на один стрибок швидше. Цю роботу можна

розпаралелювати, поділивши перебір адресного простору на декілька ділянок, і перебирати кожну з них на своїй машині.

У процесі побудови топології мережі можливо зіткнутися з випадком, коли два шлюзи налагоджено так, що віддають пакет один одному. І пакет крутиться між ними, доти, поки не вичерпає свого TTL. Це наслідок помилки системного адміністратора. У мережі такі помилки призводять до неможливості проходження пакета між двома комп'ютерами. При виявленні такої помилки користувачеві потрібно зв'язатися з системним адміністратором, і той усуне помилку. У нашому університеті обмін даними між комп'ютерами різних кафедр нечастий. Тому помилка такого роду нікому не заважає, видна лише на заняттях, подібних цьому, і може існувати в мережі роками.

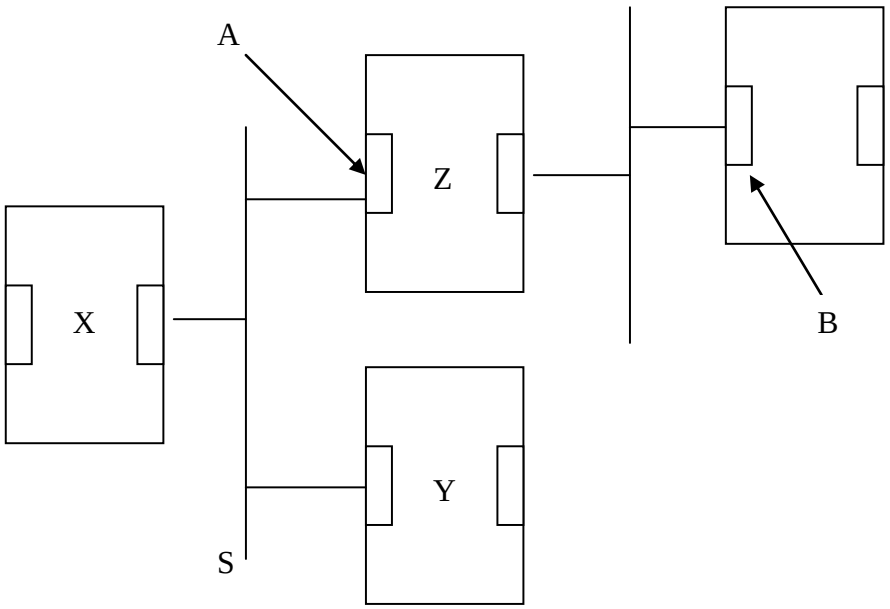


Рисунок 3.5 – Некритична помилка системного адміністратора

Інший цікавий випадок показаний на рисунку 3.5. У цьому випадку пакет, посланий за адресою B, приходить по шлюзові X, потім передається по сегменту S на шлюз Y. І звідти передається по цьому ж сегменту S на шлюз Z. Така помилка системного адміністратора не призводить до

недосяжності між комп’ютерами, але подовжує деякі маршрути на один стрибок.

У наших експериментах це виявляється в тому, що адреса A досягається в K стрибків, а адреса B в K+2 стрибка.

Для виконання цієї лабораторної роботи вам необхідно знати адресний простір, в якому знаходяться машини університету. Використовуються три інтервали:

a) 172.16.0.0 – 172.17.255.255

b) 194.44.235.0 – 194.44.235.255

c) 192.168.*.* – 192.168.*.*

Посилаючи сигнал на різні адреси цього інтервалу і аналізуючи результати експериментів, ви можете визначити працюючі комп’ютери і сегменти мережі, яким вони належать, і зв’язок між сегментами. Топологія мережі не залишається незмінною, оскільки весь час комп’ютери і шлюзи включають і вимикають. Усього в мережі університету більше 50 сегментів. Ваше завдання за цю пару знайти не менше 8 сегментів та 2–3 комп’ютери на кожному з цих сегментів. Рисунок, який ви отримаєте в результаті цієї роботи, буде вихідними даними для наступної лабораторної роботи.

3.2. Програма виконання роботи

1. Перевірити вказані діапазони комп’ютерних адрес і розбити їх на сегменти. Знайти діапазони адрес, які не використовуються.

2. Навчитися знаходити межи сегментів.

3. Навчитися знаходити задні мережні карти комп’ютерів.

4. Знайти 8 різних сегментів мережі університету, в кожному сегменті по 2–3 різних комп’ютери.

5. Нарисувати отриману топологію мережі на подвійному аркуші в клітинку. Показати, як сегменти зв’язані між собою.

Контрольні запитання

1. Які переваги в заданні інтервалу адресів за допомогою початкової адреси маски.

2. Який експеримент дозволяє перевірити, що адреса є початковою адресою сегмента.

3. Яка реакція буде на звернення за допомогою tracert до кінцевої адреси сегмента, в якому знаходиться користувач? А до кінцевої адреси іншого сегмента? Чим викликана різниця?

4. За допомогою якого експерименту можна виділити задню мережну карту комп'ютера?

5. Поясніть, чому топологія мережі, яка побудована на одній парі, відрізнятиметься від топології, побудованої на іншій парі?

6. Чим пояснюється експеримент, в якому пакет посланий від tracert «бігає» між двома комп'ютерами?

Лабораторна робота 4

ПОБУДОВА ТАБЛИЦІ МАРШРУТИЗАЦІЇ

Мета роботи: навчитися виявляти роутингову таблицю шлюзу за допомогою команди tracert.

4.1. Загальні відомості

На цій лабораторній роботі ми вирішуватимемо задачу «чорного ящика». Тобто існує об'єкт (сама таблиця маршрутизації), побудований за певними відомими правилами, але з невідомим внутрішнім вмістом. За допомогою команд traceroute, що посилаються на різні цільові адреси, пропускатимемо через нього тестові сигнали. І на підставі поведінки цих сигналів на виході робитимемо висновки про вміст досліджуваного об'єкта.

Таблиця маршрутизації складається із записів, кожна з яких описує правила поведінки з однією адресою, адресами одного сегмента мережі і адресами групи сегментів. При приході на комп'ютер пакета, він перевіряється на відповідність рядкам таблиці маршрутизації. Перевірка йде зверху вниз. Якщо знайдена відповідність рядку N, то правило рядка N визначає, що потрібно зробити з цим пакетом. І рядки, розташовані в таблиці пізніше, можна не розглядати.

У таблиці маршрутизації рядок описується 4 мя основними числами. Перших два числа описують множину адрес, яка підпорядкована цього рядка. Наступні дві адреси описують, що потрібно робити з пакетом, який відповідає правилу цього рядка.

Як можна задати множину IP–адрес, відповідних рядку? Найпростіший спосіб – вказати початкову і кінцеву адреси. Цей спосіб зручний для людині. Для обчислення відповідності або невідповідності адреси множині, заданому у такий спосіб, потрібно виконати два порівняння. Є більш економічний з точки зору машинного часу потрібного для перевірки, спосіб задання інтервалу адрес. У цьому варіанті інтервал задається двома числами – початковою адресою, званою destination, і мережною маскою (netmask). Для перевірки відповідності адреси x рядку таблиці маршрутизації необхідно скористатися формулою (x and netmask) = destination. Якщо умова виконується, то пакет, адреса призначення якого x, підпорядковується правилу рядка. Якщо умова не виконується, то пакет перевіряється на відповідність наступним рядкам таблиці.

Мережна маска – це число з 32 біт (збігається за довжиною з IP адресою). Воно записується як і IP адреса, що містить 4 байти. У вигляді чотирьох чисел від 0 до 255, розділених між собою крапками. Але маска може набувати не всіх значень. Для того щоб маскою задавався безперервний інтервал адрес, вона повинна мати вигляд послідовності одиничних біт, за якими ідуть нульові біти. У масці не може бути нульового біта, за яким йде одиничний. В результаті такого способу задання маска дозволяє задавати

інтервали, в яких кількість адрес є міра двійки. Тобто можна задати інтервали з 1, 2, 4, 8, 16, 32, 64, 128, 256 і так далі адрес. Інтервал, що містить іншу кількість адрес, за допомогою маски задати не можна, і тому у всіх сегментах мережі кількість адрес повинна збігатися з числом з цього ряду.

Є швидкий спосіб підрахувати маску, якщо відома кількість адрес описуваного нею інтервалу. Хай потрібно підрахувати маску для k адрес. Тоді порівнюємо k і 256. Якщо k менше або рівне 256, то маска буде 255.255.255.(256-k). Якщо k більше 256, то ділимо її на 256, останній байт маски 0, а результат ділення k1=k/256 поміщається в попередній байт. Маска набере вигляду 255.255.(256-k1).0. Якщо і k1 більше ніж 256, то продовжуємо ділити його на 256 і зрушувати в попередні байти маски.F

Початкова адреса інтервалу, заданого за допомогою маски, може набувати не всіх значень, а лише ти, які діляться на кількість адрес в інтервалі без остачі. Іншими словами, що в destination в кінці повинно бути не менше нульових біт, чим в netmask, відповідній їй.

Після того як за допомогою destination та netmask знайдений рядок, відповідний адресі призначення пакету, два числа, що залишилися, задають правило, що з цим пакетом робити. Одне з чисел, зване interface, задає IP мережної карти, яка використовується для передачі пакета в мережу. Воно також задає сегмент, до якого ця мережна карта приєднана і в яку пакет буде переданий. Друге число, зване gateway, є IP–адресою шлюзу, якому пакет потрібно віддати. Але ця IP–адреса завжди є адресою мережної карти, розташованої на одному сегменті з interface. Тому передачу пакета за допомогою цієї адреси можна здійснити в один стрибок та використовуючи ARP і Ethernet протоколи. На наступному комп’ютері, на який попаде пакет, буде своя таблиця маршрутизації. Буде виконана така ж перевірка, куди пакет передавати далі. При правильно налагоджених таблицях маршрутизації в результаті кожної такої передачі пакет наближатиметься до комп’ютера, чия IP адреса вказана в його полі призначення. Поки не досягне цільового комп’ютера.

Перечислимо, з яких рядків складається таблиця маршрутизації. У ній є по одному рядку для кожної мережної карти комп'ютера. По одному рядку для кожного сегмента, прилеглого до мережних карт. Число таких сегментів збігатиметься з числом мережних карт. Крім того, в таблиці будуть один або декілька рядків для кожної машини, яка може послужити проміжним шлюзом для передачі пакета в іншу частину мережі. Таких шлюзів може бути один або декілька.

Порахуємо, скільки рядків буде в таблиці маршрутизації комп'ютера з адресою 172.16.201.1 – шлюзу, що пов'язує нашу кафедру із зовнішнім світом. Для цього скористаємося топологією мережі університету, яку ми нарисували на минулій лабораторній роботі. Видно, що в цієї машини дві мережні карти. Їх IP адреси нам відомі. Кожна карта з'єднує комп'ютер зі своїм сегментом. Один сегмент містить машини нашої кафедри, а другий сегмент з'єднує комп'ютери різних кафедр корпусу У/2. Всі команди traceroute, що проходять через досліджувану машину і, досягаючи цілі більш ніж за два стрибки, як адреса що йде за 172.16.201.1 показують адресу 172.17.56.1. Інших шлюзів, з'єднаних з досліджуваною машиною на рисунку, що відображає топологію мережі, немає. Ми знайшли 5 рядків. По дві для мережних карт і для прилеглих сегментів і один для шлюзу-посередника.

Рядки, що описують мережні карти, задають множину з однієї адреси. Тому destination у таких рядках збігатиметься з IP адресою описуваної мережної карти, а netmask дорівнюватиме 255.255.255.255. Число в стовпці gateway буде 0.0.0.0, що означає, що посередників для передачі пакету до місця призначення використовувати не треба. У полі interface знаходитиметься адреса самої мережної карти.

Рядки, які задають прилеглі сегменти, містять початкову адресу сегмента в полі destination і маску, пораховану на підставі кількості машин у сегменті, за правилами описаними вище. У полі gateway в них розташоване 0.0.0.0, тому що посередники для передачі пакета до цільового комп'ютера не потрібні. І можна передавати пакет відразу за адресою з поля призначення

в один стрибок за допомогою Ethernet–протоколу. У полі interface вказується адреса тієї мережної карти комп'ютера, яка пов'язана з прилеглим сегментом.

Для рядків gateway, що описують, в полях destination та netmask задається опис множини пакетів, відповідних до цього правила. У полі gateway знаходиться IP адреса мережної карти, якій потрібно віддати пакет.

У кожній таблиці маршрутизації є спеціальний рядок – gateway за умовчанням. Його завдання – задавати правило для всіх пакетів, які не були описані в попередніх рядках. Тому він завжди розташовується останнім. У ньому destination і netmask дорівнюють 0.0.0.0, і він задає універсальну множину, в яку входять всі IP–адреси. Як gateway в такому рядку вказується адреса машини, якою потрібно віддавати більшість пакетів. Звичайно це машина, яка пов'язує поточний шлюз з основною частиною Інтернет. У Інтернеті мільйони сегментів, і описати кожен з них не вийде. Але рядок за умовчанням дає можливість описати їх як одну множину.

Розглянемо машину з адресою 172.17.56.1. Від попередньої вона відрізняється тим, що у неї не один, а два gateway. Один веде в університетську мережу, а другий – зв'язує з сегментом нашої кафедри. Тому в її таблиці маршрутизації буде на один рядок більше. У цьому рядку буде описано множину машин нашої кафедри. При цьому destination і netmask не зміняться в порівнянні з тим, як цей сегмент був описаний в таблиці маршрутизації машини 172.16.201.1. А як gateway буде використана адреса мережної карти, яка на цьому шлюзі розташована ближче до досліджуваної машини.

4.2. Програма виконання роботи

1. Під керівництвом викладача побудувати роутингову таблицю для шлюзу кафедри

1.1. Розрахувати кількість рядків у таблиці.

1.2. Заповнити рядки, які відповідають за мережні карти комп'ютера.

1.3. Виконати експерименти, які дозволять обчислити інтервали адрес, які використовуються в сегментах, що прилягають до комп'ютера.

1.4. Провести експеримент, що дозволяє визначити «дальню» мережну карту комп'ютера.

1.5. За топологічною схемою, побудованою на третій лабораторній роботі, знайти адреси мережевих карт, які використовуються як gateway для досліджуваного шлюзу. Побудувати описують рядки, які описують їх у таблиці маршрутизації.

1.6. Побудувати рядок, який описує gateway за замовчуванням.

2. Побудувати роутингову таблицю для шлюзу, що з'єднує корпус У2 з мережею університету. Побудова проводиться за аналогією зі шлюзом кафедри. У таблиці необхідно врахувати, що в шлюзі корпусу повинен бути рядок, який описує мережу кафедри.

Контрольні запитання

1. Які експерименти дозволяють знайти задню карту комп'ютера?

2. Навіщо в рейтинговій таблиці потрібен рядок з gateway за замовчуванням?

3. Як знаходяться нижня і верхня межі інтервалу адрес, які виділені для сегмента?

4. Який висновок можна зробити, якщо при запуску програми tracert на певну адресу видаються рядки, які містять три зірки замість часу кругового обігу?

Лабораторна робота 5

МОДЕЛЬ КЛІЄНТ – СЕРВЕР

Мета роботи : навчитися працювати з програмами, що реалізують протоколи telnet та ftp.

5.1. Загальні відомості

Для обміну даними по мережі повинен виконуватися обмін даними між двома програмами. При цьому на початку обміну ці програми не є рівноправними. Одна програма, яка називається сервером, надає послугу, а друга, яка запитує послугу і отримує її, називається клієнтом. Програми клієнт і сервер можуть перебувати як на одному комп'ютері, так і на різних комп'ютерах, зв'язаних мережею. Мережа, яка реалізує засоби, при цьому є середовищем, через яке ці програми взаємодіють між собою. Комп'ютери, на яких працюють ці програми, також можна назвати клієнтом і сервером. Але первинна при цьому роль програм.

Програма клієнт, виконання якої почалося у вашій системі, коли запущений запит на зв'язок по мережі, повинна:

1. Установити мережне з'єднання з сервером за допомогою протоколу TCP.
2. Прийняти від користувача вхідні дані в зручній для нього формі.
3. Перетворити ці вхідні дані в стандартну форму для передачі по мережі і послати їх серверу.
4. Прийняти від сервера вхідні дані в мережному форматі.
5. Переформатувати отримані дані для відображення на екрані вашого терміналу.

Мета роботи: навчитися передавати дані по мережі між програмами на різних комп'ютерах за допомогою стандарту сокетів.

Програма сервер виконується на комп'ютері, що надає послугу. Якщо програма сервер не працює, то запитана клієнтом послуга не доступна. Послуги надають спеціальні програми, які у WINDOWS називають резидентними, а в UNIX – демонами. Коли до них немає звернення, вони

знаходяться в пам’яті і нічого не роблять. При зверненні вони виконують покладені на них функції.

Якщо програма сервер готова приймати запити, то вона виконує такі дії:

1. Інформує мережне програмне забезпечення про те, що вона готова до встановлення з’єднувань.

2. Чекає запиту в стандартній формі.

3. Обслуговує цей запит.

4. Посилає результати назад програмі клієнту у стандартному форматі.

5. Очікує наступний запит.

В Internet взаємодіють сервера і клієнт, що виконують різні функції. Однією з функцій зручних при використанні мереж є робота на віддаленому комп’ютері. У цьому випадку користувач, сидячи за екраном і клавіатурою одного комп’ютера, використовує процесор, пам’ять, зовнішні пристрої (відмінні від тих, які застосовуються як інтерфейс) іншого комп’ютера. Таку взаємодію сервера і клієнта називають віддаленим доступом. Віддалений доступ до іншого комп’ютера може містити або не містити графічні елементи. Але програми, що дозволяють графічний віддалений доступ, не стандартизовані щодо обладнання та операційної системи. Для передачі даних у текстовій формі є програми, загальні для всіх операційних систем і будь-яких пристроїв ввіду і вивода. Протокол, за яким ці програми працюють, називається telnet. Назва telnet використовується і для найпоширенішої версії програми клієнта, яка виконує функції неграфічного віддаленого доступу. Клієнт telnet – це проста програма з текстовим інтерфейсом, що дозволяє підключатися до іншого комп’ютера через Інтернет. Якщо власник або адміністратор іншого комп’ютера дав вам право підключатися до нього, програма telnet дозволить вам вводити команди для доступу до програм і служб, які знаходяться на віддаленому комп’ютері, як ніби ви працюєте безпосередньо за ним. Клієнт telnet можна використовувати для різних завдань, у тому числі для доступу до електронної пошти, баз

даних або файлів. Стандартний клієнт telnet доступний для використання з WINDOWS або UNIX простим викликом програми з командного рядка. Крім нього, використовуються інші клієнтські програми, з більш зручним інтерфейсом. Наприклад, програма putty.

Можливий варіант, коли на одному комп'ютері працюють кілька програм серверів. У цьому випадку потрібно вміти визначати, якому з серверів віддати пакет, який прийшов на комп'ютер. Для раз в'язання цієї проблеми використовують поняття «порт». Це не технічний пристрій, а слово (два байти) в пакеті. Ці два байти служать селектором програми, якій пакет призначений. Пакет буде відісланий тій програмі сервера, яка відповідає за порт, заданий в пакеті. Порти, за які відповідають сервери, не перетинаються. Для установлення зв'язку між клієнтом і сервером вони повинні використовувати загальний порт. Це досягається за рахунок узгодження між програмістами, які пишуть програми клієнт і сервер. Вони вибирають один і той же порт. Але необхідно, щоб цей порт збігався з портами інших програм. У цілому це питання не стандартизоване, і бувають накладки. Але за найчастіше використовуваними програмами закріплені стандартні порти. Так, програми протоколу telnet за замовчуванням використовують порт 23. Список портів за замовчуванням можна знайти в UNIX у файлі /etc/services. А в WINDOWS у файлі c:\windows\system32\drivers\etc\services. Програми протоколу telnet можна використовувати для звернення до портів, відмінних від стандартного порту telnet. Якщо відомі команди протоколу, до якого звертаються за допомогою telnet, то з цим протоколом можна працювати, використовуючи клієнт telnet. Хоча й у менш зручній формі, ніж використовуючи клієнти цих протоколів. Ще одна можливість, що виникає при використанні telnet з нестандартними портами, це можливість перевірки, які сервери активні на комп'ютері, з яким намагаємося встановити зв'язок.

Інша важлива функція – мереж це передача файлів між комп'ютерами. Для цієї функції використовується стандартний протокол FTP (File Transfer Protocol). FTP дозволяє підключатися до серверів цього протоколу і

переглядати вміст каталогів, завантажувати файли з сервера або на сервер. Формально це щось на зразок підключення до якоїсь папки, яка знаходиться на іншому комп'ютері/сервері, використовуючи мережу або Інтернет. У випадку, якщо передача файлу була перервана з яких-небудь причин, протокол передбачає продовження підкачки файлу, що буває дуже зручно при передачі великих файлів.

Для роботи з протоколом FTP можна використовувати стандартний клієнт командного рядка. Він викликається командою ftp під різними операційними системами. Для зручності користувачів існує багато клієнтських над налаштувань над ftp. Їх загальний принцип полягає в тому, що маніпуляції з файлами, що знаходяться на різних комп'ютерах, з'єднаних мережею, відбуваються так само, як маніпуляції за допомогою звичайних файл–менеджерів з файлами одного комп'ютера. Дії користувача в такій оболонці перетворяться в стандартні команди ftp перед передачею в мережу. При роботі з FTP потрібно пам'ятати, що в передачі даних використовуються два комп'ютери: локальний, на якому запущений клієнт, і віддалений, на якому працює сервер.

5.2. Завдання на роботу

1. У провіднику вибрати пункт меню «Мережа». У ньому знайти папку public\utilites\development\putty. запустити програму putty.exe
2. Встановити зв'язок з комп'ютером 172.16.201.1, використовуючи логін stud1, stud2 ... stud6. З паролем stud. Цей комп'ютер працює під ОС UNIX.
3. Виконати кілька команд на віддаленому комп'ютері. Наприклад ls, ps, ls -a. Переконатися, що команди виконуються, використовуючи пам'ять і процесор віддаленої машини. А локальна машина служить тільки для введення команд і відображення результатів на клавіатурі.
4. Навчитися вести протокол сеансу роботи. У telnet ця функція називається лог. Для цього вибрати в клієнті putty лівий верхній кут мишкою.

У меню вибрати пункт «Change Settings», а в ньому пункт «Logging». Встановити запис всіх дій у файл і вибрати шлях до файлу в папці, в якій у вас є права доступу. Подати декілька команд. Після цього закрити сеанс зв'язку з віддаленим комп'ютером. Знайти файл, який вийшов на локальному комп'ютері.

5. Створити новий сеанс зв'язку. На віддаленому комп'ютері запустити програму telnet і з її допомогою увійти на віддалений комп'ютер вдруге. Тепер у нас відкриті два сеанси зв'язку. Перший сеанс з'єднує наш локальний комп'ютер (A) з віддаленим комп'ютером (B). Другий сеанс з'єднує віддалений комп'ютер B з іншим віддаленим комп'ютером C. При цьому в другому сеансі B виступає як клієнт, а C як сервер. Для зручності експерименту у нас як B і C використовується один комп'ютер. Але при цьому відкрито два сеанси зв'язку. Програма telnet запущена на B має текстовий інтерфейс. В основному режимі її робота відповідає протоколу telnet і збігається з putty. Якщо нам потрібно змінити налаштування програми, то перехід в керуючий режим здійснюється натисканням на клавіатурі спеціального символа, званого escape-символом. За умовчанням як escape символ використовується «Ctrl+]». Перейшовши в керуючий режим, можна набрати символ «?» і отримати підказку, які команди доступні в цьому режимі. Більш детальну підказку можна отримати, вводячи ім'я команди і знак питання після неї. Знайдіть, як у цьому клієнті виконати створення лога.

6. Якщо маємо кілька транзитивних сеансів з використанням програми telnet, то необхідно вміти управляти кожним з цих сеансів. Для цього використовується заміна escape символів на інші, відмінні від стандартного символу. Якщо escape символ у кожному сеансі зв'язку унікальний, то проблем з вибором сеансу, якому призначена команда управління, не виникає. У telnet для зміни escape символа потрібно перейти в керуючий режим за допомогою старого escape символа, а потім подати команду «set escape Ctrl+[». Можна використовувати й інші символи. Але

вибирати escape символ потрібно так, щоб він не використовувався в сеансі зв'язку і не заважав звичайній роботі в клієнті.

7. Закрити всі сеанси зв'язку. Запустити putty. Крім можливості задання віддаленого комп'ютера, клієнти telnet також дають можливість поставити порт, за яким буде встановлено зв'язок. Для віддаленого доступу за замовчуванням використовується порт 23. Вам потрібно навчитися користуватися портами, відмінними від стандартного. В клієнті putty для цього використовується поле введення поряд з полем вводу IP адреси. В клієнті telnet поле порту вводиться після IP адреси через пробіл. Встановлюючи сеанси зв'язку з різними серверами кафедри з використанням різних портів, потрібно навчитися визначати, на яких машинах запущені серверні програми.

Перейдемо до вивчення ftp. Ftp сервер встановлений за адресою 172.16.201.200 і для зв'язку з ним можна використовувати ім'я і пароль, які збігаються з тими, що ви використовували для telnet.

1. Встановити сеанс з'єднання з сервером з командного рядка ftp 172.16.201.200.

2. За допомогою команди help подивитися список доступних команд.

3. За допомогою команд get і put навчитися переміщати файли між локальним і віддаленим комп'ютерами.

4. Навчитися переміщати, передавати по мережі виконавчі файли, для цього передати здійсненний файл у двійковому і текстовому режимі. Запуском файлу перевірити, який режим необхідний для передачі двійкових файлів.

5. Використовуючи команди mput і mget переслати всі файли з директорії. Навчитися користуватися командою prompt.

6. Закрити сеанс ftp за допомогою команди quit.

7. УВстановити зв'язок з сервером ftp за адресою 172.16.20.5 з використанням анонімного ftp.

Контрольні запитання

1. Яка з програм сервер або клієнт починає з'єднання?

2. Назвіть кілька стандартних портів програм, відмінних від telnet.

3. Для чого застосовується протокол сеансу роботи?

4. Що таке транзитивне віддалене підключення?

5. Для чого використовується escape символ?

6. Який режим необхідний для передачі по мережі файлів, які містять рисунки?

7. У чому різниця між командами роботи з файловою системою cd і lcd?

8. Поясніть різницю роботи між командами put і get.

9. Для чого використовується анонімний ftp?

10. Які переваги у множинної пересилці файлів?

Лабораторна робота 6

ЕЛЕКТРОННА ПОШТА

Мета роботи : навчитися працювати з електронною поштою.

6.1. Загальні відомості

Електронна пошта (e-mail, від electronic mail) – технологія і надані нею послуги з пересилання і отримання електронних повідомлень (званих «листи» або «електронні листи») розподіленою (в тому числі глобальною) комп'ютерною мережею.

Електронна пошта за складом елементів та принципом роботи практично повторює систему звичайної (паперової) пошти, запозичуючи як терміни (пошта, лист, конверт, вкладення, ящик, доставка та інші), так і характерні особливості – простоту використання, затримки передачі

повідомлень, достатню надійність і в той же час відсутність гарантії доставки.

Переваги електронної пошти: легко сприймаються і запам'ятовуються людиною адреси вигляду username@domainname (наприклад, somebody@example.com); можливість передачі як простого тексту, так і тексту, який має форматну структуру, а також довільних файлів; незалежність серверів (у загальному випадку вони звертаються один до одного безпосередньо); досить висока надійність доставки повідомлення; простота використання людиною і програмами.

Недоліки електронної пошти: наявність такого явища, як спам (масові рекламні та вірусні розсилки); теоретична неможливість гарантованої доставки конкретного листа; можливі затримки доставки повідомлення (до декількох діб); обмеження на розмір одного повідомлення і на загальний розмір повідомлень у поштовій скриньці (персональні для користувачів).

Загальноприйнятим у світі протоколом обміну електронною поштою є SMTP (англ. Simple mail transfer protocol – простий протокол передачі пошти). Він використовує DNS для визначення правил пересилання пошти. DNS дозволяє вказати як приймаючий сервер будь-який вузол Інтернету. Це може використовуватися для налаштування релеінга (пересилання) пошти через треті сервери.

6.2. Задання на роботу

1. Надіслати лист самому собі з метою перевірки функціонування пошти.

2. Встановити текстовий режим надсилання листа.

3. Установити правила маркування тексту, який містився в листі, на який буде дано відповідь.

4. Розбитися на пари. Надіслати лист своєму сусідові по парі. Лист повинен містити в собі три питання, кожне питання має бути розташоване в новому рядку. Отримавши листа від сусіда, необхідно відповісти на питання.

При цьому відповідь на кожне питання розташовувати під поміченим текстом питання. Кілька разів обмінятися листами, з уточненнями відповідей з попереднього пункту. Розібратися з поняттям тред повідомлень.

5. Надіслати сусідові лист із вкладеним рисунком.

6. Надіслати лист із вкладеним виконуваним файлом. Розібратися з проблемами, які при цьому виникають.

7. Створити список розсилки з усіх студентів вашої групи. Надіслати лист за допомогою списку розсилки.

8. Надіслати лист зі звітом про виконану роботу за e-mail адресою, яка вказана викладачем.

Контрольні запитання

1. У чому перевага автоматичної позначки тексту, який був в початковому листі, при відповіді на лист?

2. Які проблеми виникають при надсиланні виконуваного файлу? Способи їх розв'язання.

3. Які переваги у списку розсилки в порівнянні з переліком множини адрес, на які має бути відправлено лист?

4. З яких частин складається адреса електронної пошти?

5. У чому різниця при пересиланні текстових і двійкових повідомлень?

6. Що таке spam?

7. Чому відсилання файлів, які можуть бути виконані, створює загрозу для приймаючої сторони?

8. За рахунок чого можлива асинхронність роботи відправника і одержувача електронної пошти?

9. Передача інформації електронною поштою вважається слабо захищеною від перехоплення. Чому?

10. Порівняйте переваги й недоліки передачі даних по e-mail зі звичайною поштою. З телефонною розмовою. З передачею інформації по Skype.

Лабораторна робота 7

ВИКОРИСТАННЯ WEB. РОБОТА ЧЕРЕЗ PROX. ІНФОРМАЦІЙНІ ПОШУКОВІ СЕРВЕРА.

Мета роботи: вивчити можливості клієнтських програм для WWW. Навчиться налаштовувати роботу через proxy. Дослідити пошук у мережі за допомогою пошукових веб-серверів і реєструвати в мережі свій сайт.

7.1 Загальні положення

Браузер, або ще так званий веб-оглядач, це спеціальна програма, яка дозволяє переглядати ті чи інші сайти. Браузер надсилає запит серверу на отримання будь-якої інформації або даних. Отримавши відповідь, інтерпретує все це спеціальним чином і показує веб-сторінку за допомогою засобів доступних на вашому засобі відображення.

Для розробки і зберігання веб-сторінок використовується мова HTML (Hyper Text Markup Language). Ця мова належить до класу SGML мов. Відмінність SGML мов у тому, що документ складається з корисної інформації, яка укладена в спеціальні «дужки» – теги. Теги описують, як інформація буде подана на пристрої, що застосовується користувачем, для відображення документа. У кожного елемента є відкриваючі і закриваючі теги, які визначають межі дії правил. У тезі, що відкриває, на подання документа впливають ім’я тега і додаткові параметри – атрибути.

Приклад:

```
<X sss = «білий»>
текст
</ X>
```

Документи на HTML називають гіпертекстом тому, що в них можна вставляти посилання на інші документи або на інші частини поточного документа. Посилання в документ вставляють за допомогою тега <A>. При цьому існує дві версії тега. Одна що використовується щоб задати аналог мітки в тексті <A name = #«ідентифікатор»>, задає якір всередині Web

сторінки. Якорем називається закладка всередині сторінки, яку можна вказати як мету посилання. При використанні посилання, яке вказує на якір, перехід відбувається не на початок сторінки, а на закладку, задану якорем. Другий варіант оператора A має вигляд <A href=«URL»>, де URL задає документ в Інтернет, до якого здійснюється перехід при виборі посилання. Перехід може бути заданий як до початку документа, так і до його частини, визначеної за допомогою першого варіанта оператора A.

URL – це єдиний указівник ресурсів (Uniform Resource Locator). Він служить стандартизованим способом запису адреси ресурсу в мережі Інтернет.

URL служить для вказівки місця розташування ресурсу в мережі. Він має таку схему:

<*схема*>: // <*логін*>: <*пароль*> @ <*хост*>: <*порт*> / <*URL-шлях*>? <*Параметри*> # <*якір*>

У цьому записі:

схема – схема звернення до ресурсу; в більшості випадків мають на увазі мережний протокол;

логін – ім’я користувача, яке використовується для доступу до ресурсу;

пароль – пароль зазначеного користувача;

хост – повністю прописане доменне ім’я хоста в системі DNS або IP-адреса хоста у формі чотирьох груп десяткових чисел, розділених точками;

порт – порт хоста для підключення;

URL-шлях – уточнююча інформація про місце знаходження ресурсу;

параметри – рядок запиту з переданими на сервер параметрами. Роздільник параметрів – знак &;

якір – описаний вище.

Проксі-сервер – служба в комп’ютерних мережах, що дозволяє клієнтам виконувати непрямі запити до інших мережних служб. Спочатку клієнт підключається до проксі-сервера і запитує який-небудь ресурс, розташований на іншому сервері. Потім проксі-сервер або підключається до вказаного сервера і отримує ресурс у нього, або повертає ресурс із власного

кешу (у випадках, якщо проксі має свій кеш). У деяких випадках запит клієнта або відповідь сервера може бути змінений проксі-сервером у певних цілях. Проксі-сервер дозволяє захищати комп'ютер клієнта від деяких мережних атак і допомагає зберігати анонімність клієнта.

Проксі-сервери застосовуються для таких цілей:

- Забезпечення доступу комп'ютерів локальної мережі до мережі Інтернет
- Кешування даних: якщо часто відбуваються звернення до одних і тих же зовнішніх ресурсів, то можна тримати їх копію на проксі-сервері і видавати за запитом, знижуючи тим самим навантаження на канал у зовнішню мережу і прискорюючи отримання клієнтом запитаної інформації.
- Захист локальної мережі від зовнішнього доступу: наприклад, можна налаштувати проксі-сервер так, що локальні комп'ютери будуть звертатися до зовнішніх ресурсів тільки через нього, а зовнішні комп'ютери не зможуть звертатися до локальних взагалі (вони «бачать» тільки проксі-сервер).
- Обмеження доступу з локальної мережі до зовнішньої. Наприклад, можна заборонити доступ до певних веб-сайтів, обмежити використання інтернету якимось локальним користувачам, встановлювати квоти на трафік або смугу пропускання, фільтрувати рекламу і віруси.
- Анонімізація доступу до різних ресурсів. Проксі-сервер може приховувати відомості про джерело запиту або користувача. В такому випадку цільової сервер бачить лише інформацію про проксі-сервер, наприклад, IP-адреса, але не має можливості визначити дійсне джерело запиту. Існують також проксі-сервери, що спотворюють, які передають неправдиву інформацію про справжнього користувача.

- Обхід обмежень доступу. Проксі-сервери популярні серед користувачів країн, де доступ до деяких ресурсів обмежений законодавчо і фільтрується.

Пошукова система – програмно-апаратний комплекс з веб-інтерфейсом, що надає можливість пошуку інформації в Інтернеті. Для пошуку інформації за допомогою пошукової системи користувач формулює запит. За запитом користувача пошукова система генерує сторінку результатів пошуку. Така пошукова видача може поєднувати різні типи файлів, наприклад: веб-сторінки, зображення, аудіофайли. Мета пошукової системи полягає в тому, щоб знаходити документи, де зазначено ключові слова, або слова будь-яким чином пов’язані з ключовими словами. Пошукова система тим краща, чим більше документів, яки релевантні запиту користувача, вона буде повертати.

Пошукові системи працюють, зберігаючи інформацію, яку вони отримують з HTML сторінок. Пошуковий робот – програма, яка автоматично проходить по всіх посиланнях, знайдених на сторінці, і виділяє їх. Робот, ґрунтуючись на посиланнях або виходячи із заздалегідь заданого списку адрес, здійснює пошук нових документів, ще не відомих пошуковій системі.

Пошукова система аналізує вміст кожної сторінки для подальшого індексування. Слова можуть бути вилучені із заголовків, тексту сторінки або спеціальних полів – метатегов. Дані про веб-сторінки зберігаються в індексній базі даних для використання в повторних запитах. Індекс дозволяє швидко знаходити інформацію за запитом користувача. Пошукова система працює з вихідними файлами, отриманими від індексатора. Пошукова система приймає запити користувачів, обробляє їх за допомогою індексу і повертає результати пошуку.

Коли користувач уводить запит у пошукову систему (зазвичай за допомогою ключових слів), система перевіряє свій індекс і видає список найбільш підходящих веб-сторінок (відсортованого за якогось критерію),

зазвичай з короткою анотацією, що містить заголовок документа і іноді частини тексту. Пошуковий індекс будується за спеціальною методикою на основі інформації, витягнутої з веб-сторінок. Більшість пошукових систем підтримує використання в запитах булевих операторів І, АБО, НЕ, що дозволяє уточнити або розширити список шуканих ключових слів. При цьому система буде шукати слова чи фрази точно так, як було введено. У деяких пошукових системах є можливість наближеного пошуку, в цьому випадку користувачі розширюють область пошуку, вказуючи відстань до ключових слів.

Корисність пошукової системи залежить від релевантності знайдених нею сторінок. Хоч мільйони веб-сторінок і можуть включати якесь слово або фразу, але одні з них можуть бути більш релевантні, популярні або авторитетні, ніж інші. Більшість пошукових систем використовує методи ранжирування, щоб вивести в початок списку «кращі» результати. Пошукові системи вирішують, які сторінки більш релевантні і в якому порядку повинні бути показані результати, по-різному. Методи пошуку, як і сам Інтернет, з часом змінюються.

7.2. Завдання на роботу

1. Запустити Internet Explorer. Завантажити будь-яку сторінку. На сторінці натиснути правою кнопкою миші і вибрати пункт «Перегляд вихідного коду». Подивитися приклад коду на HTML

2. Створити просту сторінку HTML з парою рядків тексту і посиланням на іншу, існуючу сторінку. Переконатися, що посилання працює.

3. Ознайомитися з вмістом пункту «Налаштування» в настройках. Подивитися закладки і розібратися, яка з закладок для чого використовується.

4. Знайти в опціях місце, в якому налаштовується доступ до proxy. Перевірити, що як proxy сервера вказано 172.17.10.2 і як порт – 3128.

Ознайомитися з більш докладними налаштуваннями proxy в браузері. Навчитися задавати сайти, для доступу до яких proxy не використовується.

5. Зайти на пошуковий сайт. Наприклад, на google.com. Сформувати запит, який містить два слова. Запам'ятати кількість відповідей, які відповідають цьому запиту. Сформувати запит, який містить ці ж два слова в подвійних лапках. Порівняти кількість відповідей, які знайдені цим запитом з першим запитом. Пояснити результат.

6. Розібратися, як на використовуваному вами пошукові сервері отримати доступ до розширеної мови запитів (advanced search). Які додаткові можливості в порівнянні зі звичайними запитами дає використання розширеного пошуку.

7. Навчитися реєструвати сайти, які створені вами, на пошукових серверах.

Контрольні запитання

1. Для яких функцій використовується web-browser?

2. У чому відмінність мови розмітки гіпертексту від звичайних мов програмування?

3. У чому відмінність гіпертексту від звичайного тексту?

4. Які переваги дає вихід в Internet через proxy? До яких обмежень приводить ця технологія?

5. Які Internet протоколи можуть працювати з використанням proxy, а які ні?

6. У чому відмінність мов простих запитів до пошукових серверів від мов розширених запитів? Чим ця різниця викликана?

7. Звідки пошукові системи одержують інформацію про нові сторінки і нові сайти, що розміщуються в Internet?

8. Чим визначається порядок, в якому розміщуються посилання на сторінки, знайдені при пошуковому запиті?

9. Для чого в URL використовується схема доступу? Який зв’язок між схемою доступу і іменами прикладних протоколів?

10. Як працюють пошукові роботи. Які заходи застосовуються, щоб пошукові роботи менше завантажували мережу?

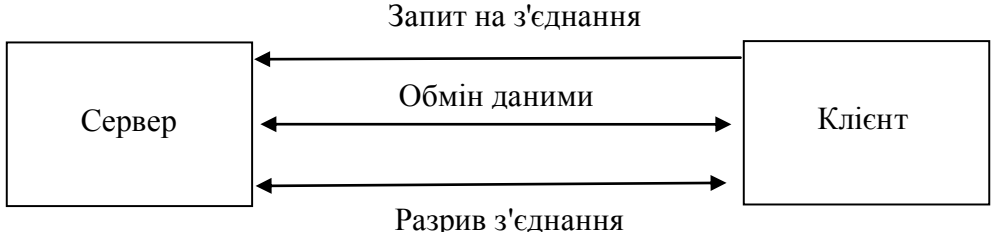
Лабораторна робота 8

РОЗРОБКА ПРОСТИХ СЕРВЕРА І КЛІЄНТА

Мета роботи: навчитися передавати дані по мережі між програмами на різних комп’ютерах за допомогою стандарту сокетів.

8.1. Загальні положення

Для передачі даних по мережі між двома машинами використовується технологія клієнт - сервер. Функціональна відмінність між клієнтом і сервером виявляється на етапі установки зв’язку між машинами і потім у процесі обміну пакетами з даними відмінність відсутня. На рис 8.1 показана схема відмінностей у функціях, які виконують програми сервера і клієнта.



Запущен весь час.

Може працювати з декількома клієнтами.

Ініціюється за запитом користувача.
Закінчує роботу по завершенню сеансу.
Пов'язаний з одним сервером.

Рисунок 8.1 –. Схема взаємодії сервера і клієнта.

Через різницю виконуваних функцій алгоритми сервера і клієнта розрізняються між собою на етапі встановлення з’єднання. І програмуються по–різному. Оскільки алгоритм роботи клієнта більш простій, почнемо з нього. На рис. 8.2 показана блок–схема виклику функцій з клієнта.

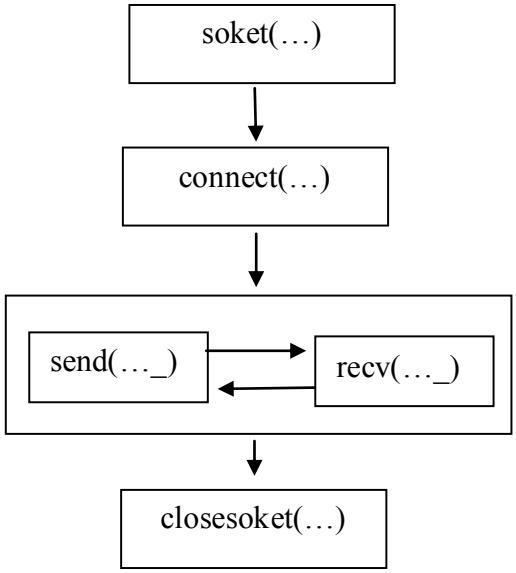


Рисунок 8.2 – Блок схема праці клієнта

Функція socket призначена для виділення місця під структури даних, в яких буде зберігатися інформація про сокеті. Її прототип:

int socket(int тип з'єднання з мережею, int тип передачі пакетів, int prt);

Функція повертає номер відкритого сокета, який буде в подальшому використовуватися у всіх функціях, які працюють з цим сокетом. Можна провести аналогію з номером файлового дескриптора, яку повертає функція open. Перший параметр задає використовуваний протокол передачі пакетів мережного рівня. Використання протоколу IP задається константою PF_INET. Інші можливі значення цього параметра в даний час, коли Internet домінує, не використовуються. Другий параметр задає використовуваний протокол транспортного рівня. Якщо використовуємо протокол TCP, то потрібно задати SOCK_STREAM. Якщо використовуємо UDP, то задаємо SOCK_DGRAM. Третій параметр задається відмінним від 0 тільки в тому випадку, якщо використовуються нестандартні пакети, задані в другому параметрі значенням SOCK_RAW. Для TCP і UDP третій параметр завжди 0. У результаті маємо виклик:

int sd=socket(PF_INET, SOCK_DGRAM,0);

Після виділення пам'яті для сокета клієнту потрібно підключитися до сервера. Це виконує функція connect з прототипом:

int connect(int sd, struct sockaddr addres, int addr_len); де sd номер сокета, який повернула функція socket, другий параметр – це структура, яка містить дані про сервер, третій параметр це кількість байт, займаних другим параметром. Перед викликом функції необхідно заповнити три поля в структурі другого параметра. Ці поля:

а) sin_family, яке відповідає за тип використовуваної адреси і для протоколу IP має значення AF_INET;

б) sin_addr, яке містить IP-адресу сервера;

в) sin_port, яке задає порт з'єднання.

Заповнення цих полів показано в прикладі програми, яка приведена нижче.

Після того як з'єднання з сервером встановлено, обмін даними здійснюється за допомогою функцій send() і recv(), порядок виклику яких визначається протоколом, який використовує програма.

Прототип функції send:

int send(int sd, char *buf, int len, unsigned int flags).

Функція повертає кількість успішно відправлених байтів. При нормальній роботі це число збігається з len. Перший параметр це використовуваний в пересиланні сокет. Другий параметр – адреса початку буфера, який містить дані, що пересилаються. Третій параметр – довжина даних, що пересилаються, у байтах. Четвертій параметр функції дозволяє змінювати роботу сокета, який використовується у функції. Якщо зміна роботи сокета не потрібна, то четвертий параметр дорівнює 0.

Прототип функції recv:

int recv(int sd, char *buf, int len, unsigned int flags).

Сенс параметрів той же, що і для send, тільки параметр len задає максимальну кількість байтів, яку програма готова прийняти. А функція повертає кількість байтів, які реально прочитані. Якщо при зверненні функція recv повернула 0, то це ознака того, що сокет був закритий.

Функція closesocket розриває з’єднання і звільняє пам’ять, виділену для функціонування сокета.

Для роботи з системою сокетів під ОС Windows її необхідно ініціалізувати. Ініціалізацію системи сокетів під Windows виконує функція WSAStartup, а звільнення пам’яті, яка виділена під систему сокетів, виконує функцію WSACleanup. Приклад роботи з цією функцією дивіться нижче.

Блок схема роботи простий програми сервера представлена на рис 8.3.

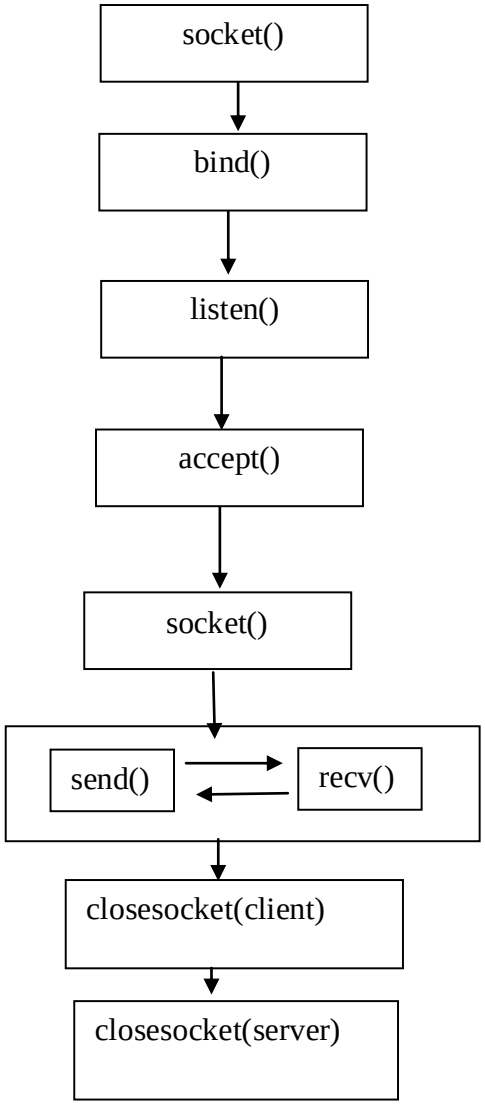


Рис 8.3 – Блок схема праці сервера

У порівнянні з клієнтом, тут відсутній виклик функції connect і є нові функції bind, listen і accept.

Функція bind виконує реєстрацію сервера у драйвера TCP/IP машини, на якій програма сервер запускається. Цією функцією сервер говорить

драйверу, що в подальшому всі пакети, які приходять на зазначений в bind порт, необхідно віддавати цьому серверу.

Прототип функції bind:

int bind(int sd, struct sockaddr *addr, int addr_len);

Другий параметр – це та ж структура, що і у функції connect для клієнта. Але на відміну від клієнта, в ній перед викликом bind потрібно заповнити тільки два поля - протокол і порт. А поле адреси заповнювати не треба, оскільки програма сервер і так знає адресу комп'ютера, на якому її запускають. Функція bind повертає код помилки. Якщо помилки не було, то повертає 0. Якщо помилка була, то код повернення функції буде відмінний від нуля, а значення помилки можна подивитися в системній зміної errno, якщо програм працює під UNIX і за допомогою функції WSAGetLastError, якщо програма працює під Windows.

Функція listen призначена для перекладу сокета, відкритого функцією socket в режим прослуховування. В цьому режимі сокет, який вказаний у формальної процедури не може бути використаний для читання або запису даних за допомогою функцій send і recv. А може бути використаний для отримання сигналів від нових клієнтів, які намагаються зв'язатися з сервером. Прототип функції listen:

int listen(int sd, int numslots), де параметр numslots показує, скільки пам'яті виділяти для клієнтів, які очікують у черзі на обслуговування. Це число рекомендується ставити від 3 до 5.

Прийом запитів на обслуговування клієнтів виконує функція accept з прототипом:

int accept(int sd, struct sockaddr *addr, int *addr_len).

Параметри цієї функції схожі на параметри функції connect, але на відміну від неї, функція accept отримує адресу і порт машини клієнта, яка встановлює з нею зв'язок. Тому структуру sockaddr перед викликом функції заповнювати не потрібно. І третій параметр передається за посиланням, оскільки він теж є параметром, що повертається. Значення, яке функція

accept повертає – це номер ще одного сокета, що відкривається для зв'язку з клієнтом. Цей сокет буде працювати в режимі читання і запису, і він відрізняється від сокета, відкритого на прослуховування. Після закінчення роботи з поточним клієнтом необхідно закрити цей сокет і знову слухати початковий сокет.

Приклад коду програми клієнта

```
#include "stdafx.h"
#include <winsock2.h>
const int BUFMAXLEN = 80;
#define WORK_PORT 10000

int main()
{
      WSAData w;
      char straddr[] = "127.0.0.1";
      struct hostent *remoteHost;
      WSAStartup(2, &w);
      int sd = socket(PF_INET, SOCK_STREAM, 0);
      sockaddr_in Service;
      Service.sin_family = AF_INET;
      Service.sin_port = htons(WORK_PORT);
      if ((Service.sin_addr.s_addr = inet_addr(straddr)) == INADDR_NONE)
      {
            remoteHost = gethostbyname(straddr);
            Service.sin_addr.s_addr =
                        *(u_long *) remoteHost->h_addr_list[0];
      }
      connect(sd,(struct sockaddr *)& Service, sizeof(Service ));
      char buf[BUFMAXLEN];
      scanf("%s", buf);
      send(sd,buf,BUFMAXLEN,0);
      closesocket(sd);
      WSACleanup();
    return 0;
}
```

Приклад коду програми сервера

```
#include "stdafx.h"
```

```
#pragma comment(lib, "Wsock32.lib")
#include <winsock2.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

const int BUFMAXLEN = 80;
#define WORK_PORT 1000

int main()
{
	WSAData w;

	WSAStartup(2, &w);
	int sd = socket(PF_INET, SOCK_STREAM, 0);

	sockaddr_in serv;
	int len;
	serv.sin_family = AF_INET;
	serv.sin_port = htons(WORK_PORT);
	if (bind(sd, (struct sockaddr *) &serv, sizeof(serv)))
	{
		int error = WSAGetLastError();
		printf("Error code %d",error);
		char c=getc(stdin);
		exit(1);
	}
	listen(sd, 3);
	int client = accept(sd, (struct sockaddr *)&serv, &len);
	char buf[BUFMAXLEN];
	int rc = recv(client, buf, BUFMAXLEN, 0);
	buf[rc] = 0;
	printf("%s\n", buf);
	closesocket(client);
	closesocket(sd);
	WSACleanup();
	return 0;
}
```

8.2 Завдання на роботу

1. Ознайомитися до кодом програми, яку дав викладач.

2. Відкомпілювати і запустити її. Переконатися в працездатності.

3. Змінити програму, щоб вона виконувала одну з таких дій:

3.1. Клієнт звертається до порту daytime (13) сервера з адресою 172.16.201.1, отримує у відповідь від сервера рядок, яка містить поточну час і дату і видавала їх на екран.

3.2. Клієнт передає серверу ім’я файлу, у відповідь сервер посилає клієнтові вміст файлу. Це вміст записується на диск.

3.3. Програми обмінюються повідомленнями по черзі. До тих пір, поки на одній зі сторін не буде введено ключове слово «exit».

3.4. Клієнт перебирає порти на заданій машині в заданому діапазоні та виводить на екран ті з портів, на яких виявлені активні програмні сервера.

Контрольні запитання

1. Чому програми сервера і клієнта мають відмінності?

2. Яка програма починає установку зв’язку при передачі даних між двома машинами в мережі?

3. Що таке мережевий порядок байтів? Як цей термін впливає на програму?

4. У чому різниця між сокетом, що задіяний на прослуховування і сокетом, що задіяний на введення та вивід?

5. Чим визначається порядок викликів функцій перекладу рядка з IP адресою і перекладу рядка з доменним іменем в двійковий код?

СПИСОК ЛИТЕРАТУРИ

1. Антонов В. М. Сучасні комп’ютерні мережі / Валерій Миколайович Антонов. – К. : МК-Прес, 2005. – 480 с.

2. Новиков Ю. Компьютеры, сети, Интернет: Энциклопедия: Наиболее полн. и подроб. рук. / Ю. Новиков, Д. Новиков, А. Черепанов, В. Чуркин; Под общ. ред. Ю. Новикова. – 2. изд. – М. [и др.]: Питер, 2003 (СПб.: ГПП Печ. Двор им. А.М. Горького). – 831 с

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторних робіт з курсу «Комп'ютерні мережі»
для студентів спеціальностей 124 – Системний аналіз,
186 – Видавництво та поліграфія

Укладач: Шахновський Юрій Сергійович

Відповідальний за випуск проф. О.С. Куценко
Роботу до видання рекомендував проф. М. І. Безменов
У авторській редакції

План 2018 р., поз. 82

Підписано до друку 11.05.2018 р. Формат 60×84 1/16.Папір офсетний.
Друк –ризографія. Гарнітура Таймс. Ум. друк. арк. 3,7.
Наклад 50 прим. Зам. № 276-18. Ціна договірна

Видавничий центр НТУ «ХПІ».
Свідоцтво про державну реєстрацію ДК№ 5478 від 21.08.2017 р.
61002, Харьків, вул. Кирпичова

Друкарня «ФОП Пісня О.В.»
Свідоцтво про державну реєстрацію ВО2 № 248750 від 13.09.2007 р.

61002, Харків, вул. Гіршмана 16а, кв. 21, тел. (057) 764–20–29